REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1996	Master's Thesis

4. TITLE AND SUBTITLE
DATA-DRIVEN PROCESS DISCOVERY
A Discrete Time Algebra
For Relational Signal Analysis

5. FUNDING NUMBERS

6. AUTHOR(S)
David M. Conrad

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P. Street
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCE/ENG/96D-01

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Steven R. LeClair
WL/MLIM
2977 P. Street, Suite 13
WPAFB, OH 45433-7746

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This research presents an autonomous and computationally tractable method for scientific process analysis, combining an iterative algorithmic search and a recognition technique to discover multivariate linear and non-linear relations within experimental data series. These resultant data-driven relations provide researchers with a potentially real-time insight into experimental process phenomena and behavior.

This method enables the efficient search of a potentially infinite space of relations within large data series to identify relations that accurately represent process phenomena. Proposed is a time series transformation that encodes and compresses real-valued data into a well defined, discrete-space of 13 primitive elements where comparative evaluation between variables is both plausible and heuristically efficient. Additionally, this research develops and demonstrates binary discrete-space operations which accurately parallel their numeric-space equivalents. These operations extend the method's utility into trivariate relational analysis, and experimental evidence is offered supporting the existence of traceable multivariate signatures of incremental order within the discrete-space that can be exploited for higher dimensional analysis by means of an iterative best-n first search.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Automated Process Discovery, Data-Driven Relational Discovery, machine learning, time series analysis, Discrete Time Series Signature, Signature-Based Abstract Algebra, Multivariate Relational Analysis, Time Series Encoding, Time Series Compression	75
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract	**PR** - Project
G - Grant	**TA** - Task
PE - Program Element	**WU** - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with....; Trans. of....; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- **DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- **DOE** - See authorities.
- **NASA** - See Handbook NHB 2200.2.
- **NTIS** - Leave blank.

Block 12b. Distribution Code.

- **DOD** - Leave blank.
- **DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- **NASA** - Leave blank.
- **NTIS** - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

AFIT/GCE/ENG/96D-01

DATA-DRIVEN PROCESS DISCOVERY:

A DISCRETE TIME ALGEBRA FOR RELATIONAL SIGNAL ANALYSIS

THESIS

David Michael Conrad
Captain, USAF

AFIT/GCE/ENG/96D-01

AFIT/GCE/ENG/96D-01

DATA-DRIVEN PROCESS DISCOVERY:

A DISCRETE TIME ALGEBRA

FOR RELATIONAL SIGNAL ANALYSIS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Computer Engineering

David Michael Conrad, B.S.

Captain, USAF

December, 1996

Acknowledgements

I first give credit and glory to God the Father for granting me this knowledge to discover something He already knew. Second only to God, I must credit my wife for granting me the strength to complete this effort. During its course, she moved our family, gave me a lovely daughter, and kept herself sane even with our three year old. My gratitude and undying love are yours always.

I want to thank my advisor, Dr. Santos, and the rest of my committee, Maj. Banks and Dr. Rogers, for their critical guidance and excellent suggestions that led to this document and the completion of my degree.

I am sincerely indebted to my sponsor, Dr. Steve LeClair, for his consistent interest, input, and motivation which kept me going through many long nights, in addition to the resources of MLIM. Specific thanks go to Dr. Sam Laube for his pulsed-laser deposition data that provided the inspirational spark for this method.

I would also like to thank Dr. Mark Oxley for his assistance in developing this abstract algebra and the mathematical notation found herein.

Lastly, I would like to thank GCS and GCE-96D for keeping morale high. Eternal thanks go to Dan Stein and Scott Brown, two exceptional Christian friends, for keeping the AFIT experience in perspective.

David Michael Conrad

Table of Contents

List of Figures

Abstract

This research presents an autonomous and computationally tractable method for scientific process analysis, combining an iterative algorithmic search and a recognition technique to discover multivariate linear and non-linear relations within experimental data series. These resultant data-driven relations provide researchers with a potentially real-time insight into experimental process phenomena and behavior.

This method enables the efficient search of a potentially infinite space of relations within large data series to identify relations that accurately represent process phenomena. Proposed is a time series transformation that encodes and compresses real-valued data into a well defined, discrete-space of 13 primitive elements where comparative evaluation between variables is both plausible and heuristically efficient. Additionally, this research develops and demonstrates binary discrete-space operations which accurately parallel their numeric-space equivalents. These operations extend the method's utility into trivariate relational analysis, and experimental evidence is offered supporting the existence of traceable multivariate signatures of incremental order within the discrete-space that can be exploited for higher dimensional analysis by means of an iterative *best-n first* search.

DATA-DRIVEN PROCESS DISCOVERY:
A DISCRETE TIME ALGEBRA FOR RELATIONAL SIGNAL ANALYSIS

I. Introduction

The term scientific discovery is generally associated with computational rather than more traditional philosophical approaches to science[1]. Generally, the discovery process "combines aspects of heuristic search in one or more problem spaces with the recognition of cues in a specific space" [21]. Up to now, most of the Artificial Intelligence (AI) 'discovery' work has emphasized one of two complementary goals[2]: the application of AI techniques to advance physical science, or the demonstration that automated search mechanisms can approximate human performance on scientific and mathematical tasks [22]. This thesis favors the former goal, presenting a comprehensive, autonomous method for signal analysis and relational scientific discovery. Specifically, this research develops an efficient search and recognition capability, within the scope of process analysis[3], to identify algebraic relations between experimental time-series variables.

Within the context of scientific process analysis, discovery is the recognition of one or more laws relating a set of observations. However, the computational discovery problem often requires searching a potentially infinite relational-space to find one relation that accurately represents the data. 'Real', noisy, erroneous, sizable, inconsistent, and/or incomplete time-series data further complicates this potentially infinite relational search [11]. Consequently, efficiency applies significantly to both search and recognition in terms of computational tractability. This research proposes an autonomous method that is capable of efficiently managing the discovery problem and is com-

[1] (Shrager & Langley 1990) provide a more thorough comparison of computational vs. philosophical science [11].
[2] Valdez-Perez cited DENDRAL (Lindsey et al. 1993) and AM (Lenat 1982) as well-known respective examples.
[3] Throughout this thesis, the term process encompasses any problem of the form of *input* $\rightarrow$ *process* $\rightarrow$ *output*.

putationally tractable, to assist researchers in the areas of signal processing, experimental data reduction, and relational process discovery.

Researchers leverage several concepts in limiting the search-space in any problem. Experience in specific domains or familiarity with analogous experiments may allow parallels to pre-existing laws as potential models, or may contribute to the efficient decomposition of complex problems. Unfortunately, domain specific knowledge is often difficult to generalize across various scientific domains. Literature, however, supports a notion that scientists tend to consider only a very limited number of functional relations to describe various processes [20]. Ideally, a tailorable search optimizes both search-space-limiting advantages.

The mathematical field of time series analysis offers many rigorous techniques to extract information from time series data. Unfortunately, the majority of these techniques either impose unrealistic assumptions on 'real' data (ie. stationary, uniformly sampled, etc.), or cannot realistically proceed in a non-exhaustive fashion. This research overcomes several of these application-limiting assumptions, exploring relational discovery from a different perspective. Interestingly, precedents exist for largely descriptive, qualitative discovery processes in fundamentally quantitative sciences [6]. The autonomous discovery method developed herein parallels such precedents, transforming real-valued series and operating over two qualitative measures.

To limit the potentially infinite relational search, this method transforms experimental time series into a well defined, discrete-space where comparative evaluation is both possible and heuristically efficient. The *discrete-space monotonicity concavity* (DMC) transform sequentially classifies real-valued data points as one of seven primitive elements[4], each representing a unique result of the cross-product of qualitative monotonicity and concavity. The encoded sequence of primitives, or more specifically, the transitions within the encoded sequence represent an equivalence class signature of the original time series. A transformed series is, therefore, represented as a sequence

[4]Hereafter referred to as primitives.

of 'primitive intervals', compressing successive occurrences of the same primitives while maintaining accurate respective durations. This interval compression often result in substantial spatial compression for smooth signals, and simplifies relational evaluation to a temporal comparison of overlapping primitive intervals across two series signatures.

The most significant aspect of this research is a template for mathematical operations inside of the transform-space that accurately parallel their numeric-space equivalents. These operations extend *DMC* into the areas of tailorable linear and non-linear trivariate analysis. Additionally, experimental evidence supports the existence of traceable multivariate signatures of incremental order within this space that can be exploited for higher dimensional analysis by means of a *best-n first* type of search.

Chapter II begins by highlighting many important concepts from time series analysis and previous AI related discovery systems, providing some background for the development of the *DMC* transform. Then, Chapter III defines *DMC*, illustrating efficient bivariate search and recognition. These ideas are then expanded in Chapter IV with the addition of transform-space binary operations, allowing trivariate discovery within the previous bivariate scope. Chapter V documents some experimental testing, and provides support for the premise of traceable signatures in multivariate relations. And lastly, Chapter VI outlines several future intentions as well as two postulated additional areas for application of these techniques, while conclusions are derived in Chapter VII.

II. Foundations of Relational Analysis

In the first chapter, scientific discovery was defined as the transition from a set of observations to one or more laws relating those observations. This chapter serves to more fully define that discovery problem, and to document previous efforts towards that end. Researchers in both mathematics and AI have proposed solutions to this problem, basing their methods upon varying combinations of search and recognition, simplifying assumptions, and domain-specific knowledge. A review of these techniques will accomplish three objectives: first, outline several hazards inherent in the problem; secondly, highlight specific weaknesses in the existing techniques; and lastly, provide some background for the *DMC* transform and its application to relational discovery, developed in subsequent chapters.

The following section refines the discovery problem, clarifying both the expected inputs and the objective. Then, Section 2.2 considers several mathematical techniques for relational analysis, while the last section highlights the lineage of relevant AI discovery systems.

2.1 Definition of the Problem

In terms of process analysis, discovery is the identification of relational laws within the context of an observed system under recognizable stimulus. Numerically, process discovery equates to the identification of a set of rational functions over the set of input variables which *surjectively* maps specific combinations of inputs onto a set of outputs. Up to this point however, consideration has not been given to the problem's domain of time series inputs. If measured time series are the basis for characterizing observed systems throughout scientific research [23], then a more precise definition is warranted.

A time series is a collection of discrete or quasi-continuous observations made sequentially in time [4]. Two properties of time series data become very important in the context of relational analysis. First, the implicit temporal ordering of successive observations allows the definition of

before and *after* relations. These two relations apply throughout any independent series, but can also generalize across multiple series in the same experiment. The second important property is that successive observations are usually not independent, and therefore, a series of this type can be exactly predicted (deterministic) or probabilistically predicted (stochastic) from past observations.

One significant hazard, when dealing with time series data, is failing to account for the temporal separation of discrete observations. When collected at uniform intervals, an individual or collection of time series can be characterized and analyzed based upon a single sampling rate. However, this by no means implies that any two sensors provide information at the same sampling rate. Likewise, the hardware responsible for collecting experimental "snapshots" can also induce irregularities, bias, or be interrupted. As subsequent sections will point out, most of the statistical techniques assume uniform sampling to their detriment.

In general, the four objectives of time series analysis are description, explanation, prediction, and/or control [4]. Descriptive analysis provides characteristic information (mean, spectrum, etc.) relative to individual time series. Explanatory analysis, on the other hand, generates information that crosses multiple series such as correlation. Prediction attempts to compute expected future observations based on the present state or values assuming either a deterministic or stochastic system. Lastly, control focuses on directing resultant system values to some pre-defined goal. These objectives are not wholly separate, but do serve to adequately classify most techniques.

2.2 Numerical Approaches

Introductory numerical analysis texts such as Mandel [15] and Chatfield [4] present a wide variety of techniques to analyze time series data. In terms of the four previously stated objectives, the process discovery problem is best categorized as explanatory analysis, attempting to recognize relations across a set of time series variables. This section introduces some basic numerical concepts as well as overviews of correlation-based, regression-based, and signature-based techniques applied

to relational discovery. These techniques, all from the domain of mathematics, either lend notional support to this research, or highlight areas of weakness that the subsequent discovery method overcomes.

Data Preparation. Transformations, of a potentially limitless variety, seem an almost basic tenant in most types of time series analysis. Generally, data transforms are applied either to recast data into an acceptable form, to perform dimensional reduction, or to temper some undesirable aspects in the data such as noise. In terms of recasting 'real' data, two primary objectives are stabilizing the variance, or imposing specific distributions, both of which strongly relate to statistical analysis [4]. Dimensional reduction, on the other hand, focuses on parsing out 'unnecessary' information, while highlighting other details. Lastly, filtering techniques, which independently represent another entire sphere of mathematics, are applied to smooth local fluctuations generally around an assumed local mean.

Filtering techniques deserve specific attention in almost any context involving 'real' data. Linear and non-linear filters represent parameterized transforms usually designed to produce output emphasizing variations at particular frequencies, while minimizing other frequencies. Choosing the appropriate filter often requires considerable experience, a knowledge of frequency aspects relative to the analysis problem and of the measurement devices involved, and a comparative understanding of the induced biases relative to specific filtering techniques [4, 9].

Of interest, relative to time series filtering, are the general equations given for common digital filters. These equations, as in Garrett [9], assume uniform spacing between successive observations, which is often an unrealistic assumption relative to experimental data. One author suggests that low-pass filtering of non-uniformly sampled data produces a separable combination of the original signal plus some additional bias [16]. Unfortunately, this separation requires a closed form equation for the original signal, which is not available in most experimental processes, and which would invalidate the need for data-driven relational discovery.

This research does not comparatively evaluate or seek to advance any one specific filter over another. It should be noted however, that low-pass filtering was used (interchangeably) with *DMC* only to demonstrate the utility of smoothing techniques to assist relational analysis and discovery.

Statistical Correlation. Correlation coefficients, cross-correlation, and cross-spectrum are three very common statistical measures that attempt to quantify the relational correspondence between two or more variables. In all three cases, the basic mechanism compares the normalized difference of each i^{th} observation from the respective series means. A relational value is then generated based on the similarity of the pattern of differences across the entire series. Regrettably, all three statistical methods are limited in their application to experimental discovery.

Correlation coefficients are cross-products of the standardized deviations of two variables with respect to their means. Three weaknesses, unfortunately, limit the application of these seemingly ideal coefficients for relational discovery. First, the constituent equation for computing numerical correlation assumes no missing values, and uniform spacing between successive points. Although there are methods such as introducing time as an independent variable or interpolating missing values, each increases the computational complexity, diminishing both the efficiency and reliability of the technique. Secondly, if $\overline{x}_i$ represents an indexed time series variable, the covariance of $\overline{x}_t$ and $\overline{x}_{t+\tau}$ can differ significantly, implying that temporal lead or lag within the process could potentially mask an input to output relation. Lastly, correlation coefficients detect only linear relationships. Assuming nonlinear relationships, recent work by Bassetti et al. have addressed this limitation by using the logarithms of variables [2].

In terms of the other two techniques, cross-correlation computationally overcomes the second previous limitation by computing the correlation coefficients between $\overline{x}_t$ and $\overline{y}_{t+\tau}$ for all τ. Intuitively, cross-correlation is therefore n times more computationally intense than its predecessor. Meanwhile, cross-spectrum adds another computational layer, applying Fourier analysis on top of the results of cross-correlation.

Unfortunately, the application of these statistical comparisons rapidly becomes too computationally time consuming to be of practical value for autonomous relational discovery.

Regression. A second possibility for mathematical process discovery is regression. Regression attempts to accurately fit predefined functions of one or more independent variable(s) to predict a single dependent variable. Often termed curve-fitting, the general approach involves tuning the parameterized coefficients of some assumed equation. Given specific coefficients and experimental values for the independent variables in question, a computational prediction of the dependent variable can then be computed for comparative evaluation.

Regression is the primary mechanism of a recent function-finding algorithm applied to experimental discovery. Chapter I introduced the premise that scientists typically consider only a very limited number of functional relations for describing a process. Citing historical records, one researcher concluded that four general functional forms account for up to 70% of all hypothesized bivariate scientific relations (*e.g.* $y = k_1 x$) [20]. The $E*$ algorithm combines regression over these four forms and statistical evaluation to fully specify equations relating two experimental variables. Testing on 217 scientific data sets[1], each containing a documented bivariate relation, demonstrated the algorithm's remarkable resolution. Although $E*$ only speculated a relation in 89 of the 217 cases, 75% of those were, in fact, correct. In comparison, other general discovery techniques often speculate an approximately equal number of incorrect relation to those correctly identified [20].

The limitations of such regressive techniques are obvious. Similarly to correlation, $E*$ only considers an extremely limited set of relations. Broader relational discovery again becomes too computationally intensive and time consuming. This thesis presents a method that automates the discovery of potential bivariate or multivariate functional forms. Potentially, those forms could then be injected into techniques such as $E*$ to refine the resolution and solve for any coefficients.

[1] Schaffer's data sets are available via anonymous ftp to ics.uci.edu from the ~/pub/machine-learning-databases directory.

Transformational Signatures. Another recent technique for relational discovery focuses on classifying linear functions based on the products, labeled equation signatures, of various transformations [7]. This approach capitalizes on post-transform similarities. Three numeric transforms (the power transforms, powers of logarithms, and exponentials of power transforms) are used to effectively produce coefficient invariant signatures for several classes of linear equations.

Although this technique is currently limited to linear equations, the basic approach is pattern recognition, and as such is only as powerful as the chosen set of features. In terms of pattern recognition, the potential growth in the number of transforms to further resolve additional forms is undefined, while the addition of any one transform may detrimentally affect any previous resolution.

The *DMC* transform is a single transform applied specifically to dimensionally reduce and represent any given linear or non-linear time series. Combinational operations (addition, multiplication, etc.) on this representation capture this method's real power for relational discovery.

2.3 Artificial Approaches

In addition to the mathematical approaches previously presented, a number of AI related systems have been developed for empirical discovery. Of those, the sequence of BACON (Langley et al. 1987) programs is generally credited as the foundation of AI related discovery systems, and as the basic reference for problem solvability. The BACON project established the continuum from data-driven to theory-driven discovery that is used for classification to this day [13]. Of interest, in terms of this research, are those systems/methods which rely on the evaluation of 'real' data, whether coupled with domain specific theory or not. This section highlights artificial data-driven discovery as demonstrated by four significant systems, including BACON.

Gerwin's Model. One discovery effort, which actually predates BACON, cognitively assesses the problem solving aspects of relational discovery. Spawning from cognitive science, Donald Gerwin set out to model human relational problem solving under experimental conditions [10].

In his experiments, test subjects were shown graphic plots of an *unknown* mathematical function with some additional random error (noise), and a base set of mathematical functions from which the unknown was related. Then, a subject was asked to specify a potential combination of base functions, which were then plotted for comparison to the unknown function. Iterations were then allowed to correct or improve any previous results.

Gerwin's work reasonably automated the basic processes employed by his test subjects. The general conclusions to emerge from this research were that extracting relations from data involves four aspects: pattern perception, classification, class specific resolution, and recycling, if necessary [10].

Unfortunately in terms of Gerwin's model, scientific research is not constrained to relations between artificial, single-variable signals. Chapter I cited some of the basic limitations of 'real' data. Incomplete information and unmeasured variables stand as a major hurdle in terms of most analysis. However, the conclusions of Gerwin's research are well taken, and all four are visible in *DMC* and the method for multivariate relational discovery developed in the proceeding chapters.

BACON.4. As previously stated, the series BACON programs are the landmark for artificially intelligent discovery. Langley cites the fourth version of the system as presenting the most complete and coherent story [14]. Being completely data-driven, BACON's basic premise is the search for 'constancy' in existing or subsequently created terms. Implementing the search for constants are three simple heuristics. The first states that if all values of a particular variable are nearly constant within a predefined threshold, then hypothesize that variable to be constant. Secondly, if one variable increases as the value of another increases, then compute their ratio (X/Y) for further examination. Lastly, if one variable increases as another decreases, then compute their product (XY).

Although seemingly obscure, these simple heuristics implement a directed exploration based on qualitative measures (similar to monotonicity). Drawing power from its ability to iteratively

generate new bivariate terms [14], BACON demonstrates the ability to rediscover an impressive set of fundamental laws from the basic physical sciences [11]. Comparisons with the regression-based system of the previous section, however, demonstrate a general tendency for BACON to spuriously presume an almost equal number of invalid relations as those it correctly discovered [20].

The discrete-space algebra developed in Chapter IV mimics BACON's ability to generate new terms. Comparably, this ability is also regarded as the major contribution of this research.

IDS. The IDS system represents a major shift along the continuum for one of the original BACON researchers. IDS specifically addresses three aspects of the discovery problem: taxonomy formation, qualitative discovery, and quantitative discovery [18]. The basic premise uses data to generate a coherent, qualitative, state-based model, retaining some numerical relations inside specific states. IDS incorporates the discovery of bounded numerical relations, similar the BACON, Abacus (Falkenhainer & Michalski 1986) and Fahrenheit (Zytkow et al. 1990), but adds a very original dimension. IDS focuses on events, conditions, etc., which cause transitions within the qualitative model, embedding relational information not only in the states, but along the transitions as well.

IDS represents significant strides for discovery and modeling. The level of symbolic information represented in the qualitative states made the IDS representation extremely readable. However, IDS partially departs from the strictly autonomous approach, requiring certain levels of interaction with human-experts. Additionally, model growth is extremely dependent on the ordering of observations, which hindered the generality of its models [18].

The level of process modeling, accomplished by IDS, is currently beyond the scope of this research. Other, very similar modeling techniques are found in the field of qualitative reasoning (see Abrams [1]). This research potentially could function as a mechanism inside such systems for autonomously discovering relational information.

KEDS. Finally, highlighting one last area, the KEDS system pairs heuristic decomposition, referred to as *split and fit*, with statistical regression. KEDS addresses an interesting domain of problems in which different relationships can hold between variables in different parts of the problem-space [19]. KEDS is a model-driven discovery system that uses mathematical relations to partition experimental data. This task of partitioning the domain space is closely linked to the expected relationships to be discovered [19]. Specifically, KEDS considers a set of parameterized polynomial functions to be fit into each partition.

Such a domain of problems with variable relations almost necessitates decomposition, however, this class is not considered in the method developed in the proceeding chapters. The term process, with regard to this thesis, is assumed to be a set of constant functions of potentially more than one variable.

III. Automating Bivariate Search and Recognition

In the previous chapter, the problem of relational discovery was formally defined and a number of mathematical and AI related techniques were subsequently presented. The origins of the transform that follows largely parallels some of the same foundational thinking as Devaney's equation signature approach (Section 2.2), but incorporates vastly different, more BACON-like, mechanisms. The basic hypothesis supporting this research can be stated as follows:

Premise 1 *Given a time series data set representing a specific experiment, observed variables (independently or in combinations) can be evaluated to identify and describe the algebraic form of multivariate relations.*

Chapter I introduced the basic representation as sequences of primitives, defined by the cross-product of monotonicity and concavity, over corresponding temporal intervals. These primitive intervals become the 'genetic sequence' or 'signature' for a given time series. These equivalence class signatures can then be compared and later combined to identify relational similarities.

The first section of this chapter presents the rationale behind the pairing of monotonicity and concavity to represent time series data. This rationale is followed by some basic time series notation in Section 3.2 that is used throughout Chapters III and IV. Section 3.3 rigorously defines the three components of the *DMC* transform, which dimensionally reduce and then compress real-valued series into sequences of primitive intervals. Thereafter, Section 3.4 documents three important properties (shift invariance, scale invariance, and negation) of the *DMC* transform. And finally, the chapter concludes with the development of a method to efficiently accomplish bivariate relational discovery.

3.1 Motivation for the Representation

The pairing of monotonicity and concavity to represent specific temporal intervals originated from a presumption about human visual processing. Our natural ability to visually observe time series waveforms, and subsequently identify patterns is astounding. The basic presumption is that

the human brain synchronizes similar periodic behavior over equivalent intervals, irrespective of scale. Figure 3.1 illustrates this notion with a real example from a materials processing technique called pulsed laser deposition (PLD[1]).

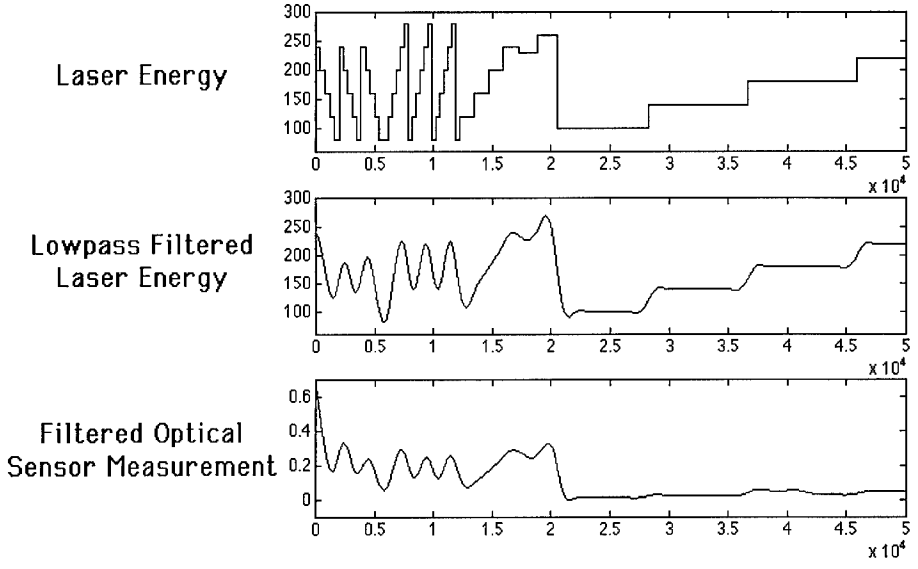


Figure 3.1 **Visual Relationship Identification.** Three actual time series signals from a PLD experiment. Visibly, all three signals appear directly related. The first signal is the input energy setting for the process laser. The second is the same signal passed through a 3^{rd} order low-pass digital Butterworth filter. The last signal is a similarly filtered optical sensor measurement of the quantity of vaporized diamond-like carbon. Therefore, in terms of process discovery, input laser energy can be hypothetically related to the quantity of a target species inside of this process.

The experimentation conducted by Donald Gerwin during the development of his system for scientific generalization (Section 2.3) lends some support to this presumption. In this case, a basic speculation about the mechanism applied by Gerwin's test subjects to accomplish the necessary relational matching has been made.

[1]The PLD process is a materials engineering thin film growth technique which uses pulsed laser radiation to vaporize materials and to deposit thin films in a vacuum chamber [5].

One critical property in terms of describing the monotonicity and concavity within a discrete time series is highlighted by Figure 3.1. This representation assumes smoothness between successive sample points. Although the definitions of monotonicity and concavity presented in Section 3.3 are insensitive to sampling rates, smooth, assumably continuous, functions allow the interpolation any number of additional data points. Undersampling or overwhelming noise naturally impedes relational discovery by compromising the accuracy of any representation.

In many instances, filtering input series appropriate to the observational sampling can effectively reduce noise and induce smoothness. Specific to this representation, filtering step functional inputs similar to "Laser Energy" in Figure 3.1 or signals containing high-frequency noise to produce continuous renderings more efficiently represent the patterns of low-frequency change relative to comparative relational evaluation. Ideally, any number of transformed renderings of experimental series can be included as input to this method at the discretion of the researcher. Realistically, however, each additional input increases the size of the search space, and consequently, the computational time of any method.

3.2 Time Series Notation

Section 2.1 presented the basic concepts of a time series as a sequence of observations. This section serves to formally specify the mathematical notation used for these concepts throughout this and the next chapter.

First, consider that every discrete observation is measured at some specific instant in time, and that any instant occurring after any other instant must be of greater value. In most cases, each time series variable, or the entire set of experimental variables are paired with a sequence of time-stamps relative to each observation. This pairing allows the following definition.

Definition: The Finite Sequence of n Observation Sample Times

$$\bar{t} = (t_1, t_2, \ldots, t_n) \quad \text{such that } t_{i+1} > t_i \tag{3.1}$$

Next, the formal specification of an observation builds on the previous definition of observation sample times. For purposes of this research, observations are simply an injective mapping, represented as the result of a unique real-valued function, of observation sample times to elements of the real numbers.

Definition: A Time Series Observation

$$\begin{gathered} a_i = \mathrm{F}_a(t_i) \quad \text{for } i = 1, 2, \ldots, n \\ \text{such that } a_i \in \Re \end{gathered} \tag{3.2}$$

Consequent to Equation 3.2, only one final notational definition remains.

Definition: A Time Series of Indexed, Observations

$$\bar{a} = (a_1, a_2, \ldots, a_n) \tag{3.3}$$

Throughout the remainder of this thesis, vectored lowercase letters imply an entire time series of n observations, while lowercase letters with an associated subscript imply a specific real-valued observation indexed by the subscript, which in turn is associated with a similarly indexed sample time.

3.3 Definition of the DMC Transform

Fundamentally, the *DMC* transform is actually a series of three numeric transformations. The first component is the qualification transform ($\mathcal{Q}_\mathrm{T}$), which computes the qualitative measures

of monotonicity and concavity. $\mathcal{Q}_T$ transforms real-valued observations into a small set of integer bivariates. Second is the encoding transform ($\mathcal{E}_T$). The $\mathcal{E}_T$ transform encodes each bivariate generated by the $\mathcal{Q}_T$ transform into the set of positive integers, effectively using one integer to represent the previous pairing of two. The last component is the compression transform ($\mathcal{C}_T$), which as the name implies, compresses intervals of repeated integers down to a single record. These records contain the corresponding encoded integer, plus two time indexes denoting the initial and terminal sample times.

Each of these component transforms will be rigorously defined in the next three sections. Then, Section 3.3.4 abstracts to the collective *DMC* transform, presenting a unified summary and illustration. Relative to the terminology introduced in Chapter I, the term *primitive* refers to the discrete values produced by the qualification and encoding transforms specifically reference a unique result of the cross-product of monotonicity and concavity. Additionally, the records generated by the compression transformation implement the concept of a *primitive interval.*

3.3.1 The Qualification Transform. A monotonic sequence implies either consistently increasing, or consistently decreasing in value. Initially, consider encoding a time series based solely on monotonic segments (increasing, constant, or decreasing). Figure 3.2 illustrates the piecewise monotonic encoding of the first 10,000 observations from the two filtered series of the PLD data originally given in Figure 3.1. Such an encoding would seem adequate to capture the periodic behavior assumed in the previous section. This example demonstrates not only a strong correlation between the laser and emission signals after the transformation, but also illustrates the potentially huge representational space savings of interval compression[2]. However, the use of only monotonicity

[2]Consider that each of the three signals depicted in Figure 3.1 are composed of 50,000 data points collected over a five hour period. Their resultant 'monotonic' encodings reduce to 43, 23, and 25 records respectively. Assuming 32-bit floating point values for each of the observation, and 32-bit integers for each of the record fields, discrete-space encoding reduces the required storage space from 600,000 byte to just 1092 bytes. However, spatial savings is considerably more important in terms of efficient computational search.

was considerably weaker in terms of realistic discrimination than the pairing of monotonicity and concavity, which on average, only doubles the size of the reduced representation.

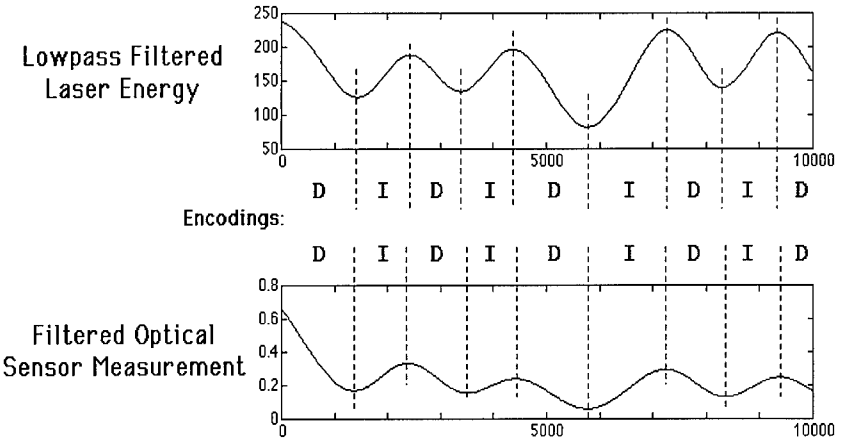


Figure 3.2 **Monotonic Encoding.** An example demonstrating the piecewise encoding of two strongly correlated segments from the original PLD data, introduced previously in Figure 3.1.

The qualitative measure of concavity, which describes the curvature of a segment, was paired with monotonicity, as described above, to enhance the representational 'signature' of any given time series. The initial choice of monotonicity defines a certain number of functional equivalence classes. The pairing of monotonicity and concavity effectively subdivides each of the monotonic equivalence classes into a much larger number of unique 'signature' classes. This additional resolution serves to improve accuracy during relational discovery, and to differentiate operational results, which are developed in the next chapter.

In many respects, the monotonicity and concavity defined for this transform mirror basic differencing techniques, which correspond to the discrete forms of the first and second derivatives, with an underlying assumption of differentiability. Qualitative monotonicity, as previously illustrated, is

characterized on the range of monotonically increasing, constant, or monotonically decreasing over the domain of a series of real numbers. Qualitative concavity, on the other hand, is represented as either convex[3], constant, or concave[4] over the same domain. Numerically, the respective ranges are simply derived from the relational operations of greater than, less than, and equal to, as shown in the following three equations.

Definition: The Qualification Transform

$$\begin{gathered}(\mathsf{M}_i,\ \mathsf{C}_i) = \mathcal{Q}_{\mathrm{T}}(a_i) \\ \text{where } \mathsf{M}_i,\ \mathsf{C}_i \in \{+1,\ 0,\ -1\} \quad \text{for each } i = 2,\ 3,\ \ldots,\ n-1\end{gathered} \tag{3.4}$$

Definition: Qualitative Monotonicity

$$\mathsf{M}_i = \begin{cases} +1 & \text{if } a_i > a_{i-1} \\ 0 & \text{if } a_i = a_{i-1} \\ -1 & \text{if } a_i < a_{i-1} \end{cases} \tag{3.5}$$

Definition: Qualitative Concavity

$$\mathsf{C}_i = \begin{cases} +1 & \text{if } \frac{a_{i+1}-a_i}{t_{i+1}-t_i} > \frac{a_i-a_{i-1}}{t_i-t_{i-1}} \\ 0 & \text{if } \frac{a_{i+1}-a_i}{t_{i+1}-t_i} = \frac{a_i-a_{i-1}}{t_i-t_{i-1}} \\ -1 & \text{if } \frac{a_{i+1}-a_i}{t_{i+1}-t_i} < \frac{a_i-a_{i-1}}{t_i-t_{i-1}} \end{cases} \tag{3.6}$$

Of note, if uniform sampling can be guaranteed across an entire time series, the denominator in Equation 3.6 always reduces to one. This reduction can be leveraged for additional computational efficiency by eliminating two unnecessary divisions.

[3]Having the property that the chord joining any two points on its graph lies above the graph [3].
[4]Having the property that the chord joining any two points on its graph lies below the graph [3].

Although conceptually, it is easy to iterate the above definitions across an entire time series, the specific notation is given by the following definition.

Definition: The Series Qualification Transform

$$
\begin{aligned}
(\overline{\mathsf{M}},\ \overline{\mathsf{C}}) &= \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) \\
\text{such that } \overline{\mathsf{M}} &= (\mathsf{M}_2,\ \mathsf{M}_3,\ \ldots,\ \mathsf{M}_{n-1}), \\
\text{and } \overline{\mathsf{C}} &= (\mathsf{C}_2,\ \mathsf{C}_3,\ \ldots,\ \mathsf{C}_{n-1})
\end{aligned}
\tag{3.7}
$$

3.3.2 The Encoding Transform. Computationally, the cross product of monotonicity and concavity would contain nine unique pairings. But given the domain and the relative meaning inherent in each pairing, it does not make sense to consider constant values with a concavity other than constant. Figure 3.3 illustrates the resultant set of seven primitives based upon the remaining (monotonicity, concavity) pairings.

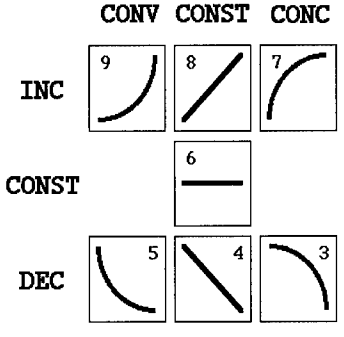


Figure 3.3 **Representational Primitives.**

For implementational simplicity, the encoding transform *injectively* maps the seven primitive elements, each representing a unique bivariate tuple, into the set of integers. The integer values

contained within each primitive outlined in Figure 3.3 demonstrate one such encoding. In this case, the specific encoding prefaces an expansion of the seven basic primitives in Chapter IV.

From the same standpoint used to define an observation, $\mathcal{E}_{\mathrm{T}}$ represents a simple function mapping bivariate tuples into the positive integers, or at this point, the Natural numbers.

Definition: The Encoding Transform

$$\mathcal{A}_i = \mathcal{E}_{\mathrm{T}}(\mathsf{M}_i,\ \mathsf{C}_i) \qquad \text{where } \mathcal{A}_i \in \mathbf{Z}^+ \quad \text{given by} \tag{3.8}$$

$(\mathsf{M}_i,\ \mathsf{C}_i)$	(+1,+1)	(+1, 0)	(+1,-1)	(0, 0)	(-1,+1)	(-1, 0)	(-1,-1)
$\mathcal{A}_i$	9	8	7	6	5	4	3

The previous definition again represents the indexed notation for single tuple encoding. The notation for encoding the entire series is given similarly to Equation 3.7.

Definition: The Series Encoding Transform

$$\bar{\mathcal{A}} = \mathcal{E}_{\overline{\mathrm{T}}}(\overline{\mathsf{M}},\ \overline{\mathsf{C}}) \qquad \text{such that } \bar{\mathcal{A}} = (\mathcal{A}_2,\ \mathcal{A}_3,\ \ldots,\ \mathcal{A}_{n-1}) \tag{3.9}$$

3.3.3 The Compression Transform. Although unnecessary, compression is a simple procedure that can significantly reduce the number of iterations (i.e. CPU operations) required for each subsequent operation, and during each evaluation. This compression defines primitive intervals starting with the first primitive, and then adding a new record whenever the primitive value changes, until the end of the sequence. The basic record denoting a specific primitive interval contains the primitive in that region, an initial time index, and the terminal time index. Simply considering that record as a 3-tuple, the definition for regional compressing to a single primitive interval is as follows.

Definition: The Compression Transform

$$\begin{aligned} &\text{if} \quad i = 2, \text{ or } \mathcal{A}_i \neq \mathcal{A}_{i-1}; \\ &\text{and} \quad j = n-1, \text{ or } \mathcal{A}_j \neq \mathcal{A}_{j+1}; \\ &\text{and} \quad \forall k \mid i < k \leq j \quad \mathcal{A}_k = \mathcal{A}_i; \\ &\quad \text{then} \quad \mathcal{C}_{\mathrm{T}}(\mathcal{A}_i, \ldots, \mathcal{A}_j) = (\mathcal{A}_i, t_i, t_j) \end{aligned} \tag{3.10}$$

Notationally, a compressed series of transformed values is denoted with a double dot, while indexed records are given a single dot and an associated subscript.

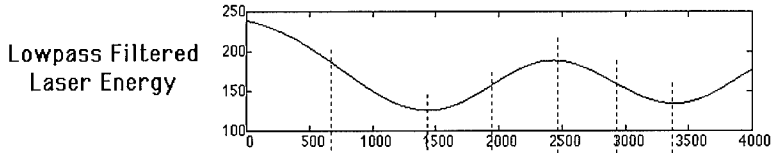
Definition: The Series Compression Transform

$$\mathcal{C}_{\bar{\mathrm{T}}}(\bar{\mathcal{A}}) = \ddot{\mathcal{A}} = (\dot{\mathcal{A}}_1, \dot{\mathcal{A}}_2, \ldots, \dot{\mathcal{A}}_{m+1}) = ((\mathcal{A}_2, t_2, t_a), (\mathcal{A}_{a+1}, t_{a+1}, t_b), \ldots), \tag{3.11}$$

where m equals the number of times the primitive value changes.

3.3.4 Summarizing DMC. The previous three sections presented the low level components to abstractly define the *DMC* transform. Collectively, the sequences of primitive intervals generated from a transformed experimental data set represent the discrete-space, equivalence class signatures of the original time series variables. Figure 3.4 illustrates the entire transformation over a small interval of one of the PLD data series originally illustrated in Figure 3.1.

The *DMC* transform makes two significant contributions to autonomous data-driven discovery. First, the transformation effectively classifies real-valued signals into a discrete-space of functional equivalence classes, which the next section distinguish as shift and scale invariant. These equivalence classes can then be compared for relational proximity. Secondly, the compressibility of an equivalence class signature often significantly reduces the computational explosion (i.e. processing time) of generic relational search. Additionally, Chapter IV enhances these two contributions by developing and demonstrating the operations of equivalence class signature addition and signature multiplication, along with a template for the development of other operations.



Primitives:

···(-1,-1)···(-1,1)···(1,1)···(1,-1)(-1,-1)(-1,1)···(1,1)···

33333 ··· 3355555 ··· 55999 ··· 9 777 ··· 7333 ··· 3555 ··· 5999 ··· 99

Primitive Intervals:

p	3	5	9	7	3	5	9
t_{Init}	1	679	1441	1954	2462	2928	3487
t_{Term}	678	1440	1953	2461	2927	3486	4000

Figure 3.4 **Primitive Interval Encoding.** The complete transformation from 4,000 real-valued observation, to 4,000 qualitative bivariate primitives, to 4,000 encoded primitives, to 7 primitive intervals.

3.4 Properties of the Transform

The *DMC* transform defined in the previous sections has three very important properties (shift invariance, scale invariance, and the operation of negation) with respect to the original search problem. The first and second properties eliminate two infinite degrees of freedom, while the last, which is actually a unary operation, provides a more precise definition of the operation, and also saves mathematical computations. These properties will each be treated in turn.

3.4.1 Shift Invariance. With respect to any given time series[5], the property of shift invariance implies insensitivity to any unilateral or bilateral translations (i.e. in arithmetic mean, in time, or in both mean and time). Each of these translations will be notationally addressed separately, with the understanding that they may be repeated and/or combined in any order.

[5]By nature, a time series is a two dimensional construct, with a presumed temporal axis.

Theorem 1 *(Arithmetic Shift Invariance) Let c denote any real-valued constant, and* **1** *denote the constant* ones *series (ie.* $\mathbf{1} = (1, 1, \ldots, 1)$*). Then, arithmetic shift invariance is given as*

$$\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a} + c\mathbf{1}) = \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a} + \bar{c}) = \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) \tag{3.12}$$

Theorem 2 *(Temporal Shift Invariance) Let* τ *denote any positive or negative time-based offset, such that* $\bar{t} + \tau\mathbf{1} = \bar{t}'$*. Then, temporal shift invariance is given as*

$$\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}') = \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) \tag{3.13}$$

The formal proof of this property requires the definition of the transform-space operation of addition (Section 4.2.1) and is given in Appendix B. Informally, monotonicity and concavity are based upon the differences between neighboring points. The addition of a constant value to the mean, and/or offsetting the specific starting time do not affect these differences however great or small the constant.

3.4.2 Scale Invariance. In contrast to shift invariance, the property of scale invariance implies insensitivity to any change in ratio between the original series and a product of the original series, where that product can be modeled as the result of a scalar multiplication by a positive constant. In terms of analysis, this property is reasonable only along the non-temporal axis of any time series.

Theorem 3 *(Scale Invariance) Let c denote any positive real-valued constant. Then, scale invariance is given as*

$$\mathcal{Q}_{\overline{\mathrm{T}}}(c\bar{a}) = \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) \qquad \forall c > 0 \tag{3.14}$$

The formal proof of this property requires the definition of the transform-space operation of multiplication (Section 4.2.2) and is also given in Appendix B. Informally, any positive change in ratio of the differences that define monotonicity and concavity will not change the aspect of those differences.

3.4.3 Discrete-Space Negation. The basic operation of negation on a time series in real-space is not precisely defined. A plausible definition could be scalar multiplication by a negative

real-valued constant. Unfortunately, this fails to address how to handle associated arithmetic shifts as previously discussed in Section 3.4.1. If the mean value of a time series is not zero, scalar multiplication by a negative constant also has the side affect of negating that mean value. In discrete-space however, the mean value is irrelevant, so simple scalar multiplication combined with the transformation becomes both an adequate and precise definition for negation.

Theorem 4 *(Negation) Let c denote any negative real-valued constant. Then, negation is given by*

$$\mathcal{Q}_{\overline{\mathrm{T}}}(c\bar{a}) = \neg\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) \qquad \forall c < 0 \tag{3.15}$$

given by the following mapping function:

$(\mathsf{M}_i, \mathsf{C}_i)$	(+1,+1)	(+1, 0)	(+1,-1)	(0, 0)	(-1,+1)	(-1, 0)	(-1,-1)
$\neg(\mathsf{M}_i, \mathsf{C}_i)$	(-1,-1)	(-1, 0)	(-1,+1)	(0, 0)	(+1,-1)	(+1, 0)	(+1,+1)

3.5 Bivariate Relational Discovery

The relatively simple mechanisms presented in Sections 3.3 and 3.4 already provide the foundation for a method capable of bivariate relational discovery. A bivariate relation maps a single hypothetical (user-defined) or actual time series variable onto another single process variable (*e.g.* $y = c_1 x$, or $y = x + c_1$). What remains is the method for the efficient search and then evaluation of candidate relations.

At this point, it is important to point out that scientific analysis focuses on any and all accurate relations, not just the first or most obvious. For that reason, this bivariate search method pairs every input series with each independent process output. The search also considers the negative image of each input paired with each output. Consequently, bivariate search is classified as exhaustive, but with the reasonable expectation that the space of bivariate relations is small when compared to the combinatorial space of higher order relations. However, further analytical development in subsequent chapters reveals one benefit of such a bivariate search. In the subsequent

chapters, the space of bivariate relations is completely spanned during the first iteration of this method for multivariate analysis.

Relational evaluation can now be addressed. In essence, evaluation involves the computation of equivalence class proximity. The basic technique traverses the temporal range of two experimental series comparing the 'primitive' signatures of one to the other. On compressed intervals, this proximity computation involves calculating specific regions of overlap and performing a single primitive comparison over that entire region. The overlapping duration can then be credited as matching or as failing to match. Therefore, in the simple bivariate case, the relational figure of merit (FOM) is defined as the sum of the durations across regions of overlap where both the monotonicity and concavity of both series are equivalent. Notationally, the concept is easier to consider on uncompressed series of primitives.

Definition: Uncompressed Bivariate Figure of Merit

$$FOM(\bar{\mathcal{A}}, \bar{\mathcal{B}}) = \frac{\sum_{i=2}^{n-1} \chi(\mathcal{A}_i, \mathcal{B}_i)(t_i - t_{i-1})}{t_{n-1} - t_1} \tag{3.16}$$

$$\text{where} \quad \chi(\mathcal{A}_i, \mathcal{B}_i) = \begin{cases} 1 & \text{if } \mathcal{A}_i = \mathcal{B}_i \\ 0 & \text{otherwise} \end{cases}$$

In terms of the FOM calculation on compressed signatures, a procedure better defines the computation.

Definition: Compressed Bivariate Figure of Merit Procedure

$I = 2$
$m_1 = 1$
$m_2 = 1$
$FOM_{Hits} = 0$
WHILE $I < (n-1)$ DO
  IF $\dot{\mathcal{A}}_{m_1}.Term < \dot{\mathcal{B}}_{m_2}.Term$ THEN
    IF $\dot{\mathcal{A}}_{m_1}.Prim = \dot{\mathcal{B}}_{m_2}.Prim$ THEN
      $FOM_{Hits} = t_{\dot{\mathcal{A}}_{m_1}.Term} - t_{I-1}$
    $m_1 = m_1 + 1$
    $I = \dot{\mathcal{A}}_{m_1}.Init$
  ELSE
    IF $\dot{\mathcal{A}}_{m_1}.Prim = \dot{\mathcal{B}}_{m_2}.Prim$ THEN
      $FOM_{Hits} = t_{\dot{\mathcal{B}}_{m_2}.Term} - t_{I-1}$
    $m_2 = m_2 + 1$
    $I = \dot{\mathcal{B}}_{m_2}.Init$
  END
END
$FOM(\ddot{\mathcal{A}}, \ddot{\mathcal{B}}) = FOM_{Hits}/(t_{n-1} - t_1)$

where $\ddot{\mathcal{A}}$ and $\ddot{\mathcal{B}}$ are two sequences of primitive intervals, and
m_1 and m_2 are respective indexes to the current record in each sequence.

Additionally, either computation for the figure of merit demonstrates the following four properties (given in compressed notation).

1. $0 \leq FOM(\ddot{\mathcal{A}}, \ddot{\mathcal{B}}) \leq 1 \quad \forall \ddot{\mathcal{A}}$ *and* $\ddot{\mathcal{B}}$
2. $FOM(\ddot{\mathcal{A}}, \ddot{\mathcal{A}}) = 1 \quad \forall \ddot{\mathcal{A}}$
3. $FOM(\ddot{\mathcal{A}}, \ddot{\mathcal{B}}) = FOM(\ddot{\mathcal{B}}, \ddot{\mathcal{A}})$
4. if $FOM(\ddot{\mathcal{A}}, \ddot{\mathcal{B}}) < 1$ then $\ddot{\mathcal{A}} \neq \ddot{\mathcal{B}}$

Finally, bivariate data-driven discovery can be modeled as the combination of the *DMC* transform, an exhaustive pairing of experimental variables, the evaluational computation of each FOM, and a final resultant sort. Figure 3.5 illustrates the basic discovery method. Optional filtering has been included for the reasons discussed in Section 3.1, along with optional regression to solve for coefficients in promising candidate relations.

Experimental results for bivariate relational discovery are given in Chapter V in conjunction with the results from higher order relational searches. But first, Chapter IV comprehensively expands this foundation to support multivariate relational discovery.

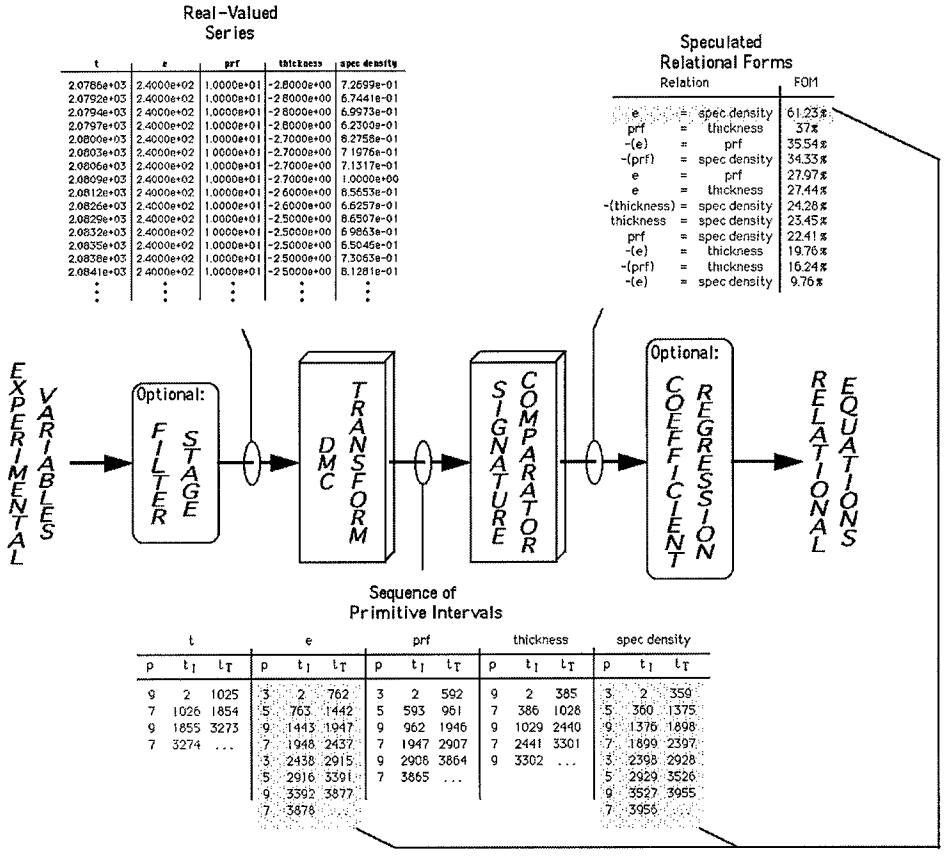


Figure 3.5 **Bivariate Relational Discovery.**

IV. Algebraic Expansion into the Multivariate

Chapter III laid the foundations for both efficient bivariate search and recognition, and a basic applicational methodology. This chapter enhances that foundation, by developing and demonstrating binary operations defined within *DMC* transform-space that parallel their numeric-space equivalents. These operations extend the method's utility into trivariate relational analysis, and experimental evidence is offered in Chapter V supporting the existence of traceable multivariate signatures of incremental order within the discrete-space that can be exploited for higher dimensional analysis by means of an iterative *best-n first* type of search.

The first section defines the notion of binary discrete-space operations, to include a discussion the potential results and the necessary extensions to the basic set of primitives in support of such operations. Section 4.2 develops the computational tables for both addition and multiplication of functional equivalence classes, consequently allowing the relational consideration of addition, subtraction, multiplication and division. Then, Section 4.3 combines these operations with the ideas of the preceding chapters into a strategic method for multivariate search and recognition, defining the mechanisms utilized in the next chapter for experimentation.

4.1 Definition of Discrete-Space Operations

The basic premise of a binary operation combines a pair of values, according to predefined rules, to produce a resultant third value related to the previous two by the operation performed. Addition and multiplication are two classic examples of mathematical binary operations, and both are developed in Section 4.2.

4.1.1 A Template for Discrete-Space Operations. With the assumption of smoothness between sample points, precise operational results are computable in numeric-space along the entire length of any two time series in question. In *DMC* transform-space however, combining two temporal regions is not necessarily guaranteed to produce only one resultant region. In several cases,

specific to a given operation, the combination of two overlapping regions results in a sequence of regions within the original overlap. In these cases, the one correct sequence of primitives and their relative durations within the overlap are dependent on the scaled, real values of the original series. However, the dimensional reduction performed by *DMC* discards the actual observational values, and any associated scales have yet to be regressed. Therefore, operations within *DMC* transform-space do not necessarily allow for the precise computation of a complete operational result.

Operations on many pairings in discrete-space result in computationally *well defined* monotonicity and concavity, while other combinations are *partially defined* in either monotonicity or concavity. At present[1], the remaining unresolvable combinations are left as *undefined* results, providing little or no useful information relative to the original task of relational discovery. Together, the well and partially defined operational results form a partial equivalence class signature, which can still be used for relational evaluation.

Equation 3.4 defined a set of three potential symbol-values for both monotonicity and concavity. To support partially and undefined operational results and maintain algebraic closure, another symbol is required to represent an unspecified series of the original three symbols. For that reason, the symbol 'u' has been added to the original set of $\{+1,\ 0,\ -1\}$ as defined below.

Definition: *DMC* Transform-Space Operational Template

$$\begin{gathered} <GenericOp>\ [\mathcal{A}_i,\ \mathcal{B}_j\ ] = (\mathcal{M}_{\mathcal{R}}, \mathcal{C}_{\mathcal{R}}) \\ \text{such that: } \mathcal{M}_{\mathcal{R}} \in \{+1,\ 0, -1,\ u\} \wedge \mathcal{C}_{\mathcal{R}} \in \{+1,\ 0, -1,\ u\} \end{gathered} \tag{4.1}$$

[1] Chapter VI proposes two possible improvements directly related to currently undefined and/or partially defined resultant regions.

With the following three classifications for operational results:

$$\begin{aligned}
&\text{(i)} \quad \text{"}Well\ Defined\text{"} \Rightarrow \mathcal{M}_{\mathcal{R}} \in \{+1,\ 0, -1\} \wedge \mathcal{C}_{\mathcal{R}} \in \{+1,\ 0, -1\}, \\
&\text{(ii)} \quad \text{"}Partially\ Defined\text{"} \Rightarrow (\mathcal{M}_{\mathcal{R}} \in \{+1,\ 0, -1\} \wedge (\mathcal{C}_{\mathcal{R}} = u))\ or \\
&\qquad\qquad ((\mathcal{M}_{\mathcal{R}} = u) \wedge \mathcal{C}_{\mathcal{R}} \in \{+1,\ 0, -1\}), \\
&\text{(iii)} \quad \text{"}Undefined\text{"} \Rightarrow (\mathcal{M}_{\mathcal{R}} = u) \wedge (\mathcal{C}_{\mathcal{R}} = u)
\end{aligned} \tag{4.2}$$

The concession allowing for undefined operational results highlights one potential shortcoming in this technique. As a minimum requirement, signals must now be sufficiently long and of sufficient variability in terms of the basic primitives such that an operational result contains enough information for adequate resolution. This idea parallels the conclusion drawn by Milosavljević concerning mutual information for jointly encoding DNA sequences [17]. In most cases, the amount of temporal data collected for scientific analysis of a process, given abilities to sample into the megahertz, is assumably adequate. Likewise, the majority of present day scientific research is not constrained to simple linear observations. Many techniques exist for manipulating exclusively linear data, almost to the point of being uninteresting. The significance of this method lies in its ability to discover linear and non-linear multivariate relations in predominantly non-linear series.

4.1.2 Updating the Bivariate Representation. The addition of undefined and partially defined operational results also mandates expanding the set of seven basic primitives from the previous chapter. The new cross product of monotonicity and concavity, including the 'unknown' symbol, produces sixteen primitives. Because Chapter III ruled out constants with a concavity other than constant, the pairing of monotonic constant with an unknown concavity can similarly be removed. Figure 4.1 illustrates the resultant set of thirteen primitives.

In terms of the three components to the *DMC* transform, the $\mathcal{Q}$ transform remains unchanged relative to this new set of primitives. The computation of monotonicity and concavity from real-values is always well defined. The encoding transformation can likewise remain unchanged because

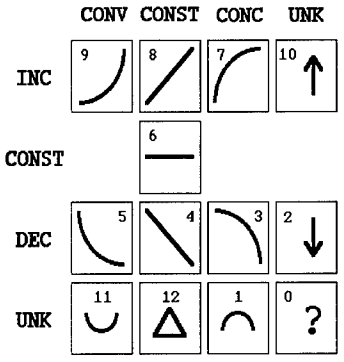


Figure 4.1 **Enhanced Representational Primitives.**

$\mathcal{Q}_T$ is providing no additional information. However, the specification of positive-integer encoding values for the new terms is given as follows.

Addendum to Equation 3.8: Encoding for Partially and Undefined Operation Results

(M_i, C_i)	$(u,0)$	$(u,+1)$	$(+1,u)$	. . .	$(-1,u)$	$(u,-1)$	(u,u)
$\mathcal{A}_i$	12	11	10	. . .	2	1	0

And finally, the $\mathcal{C}$ transform also remains unchanged. The equality test for compression realistically applies to all integers from $-\infty$ to $+\infty$.

As with the transforms, the properties of shift and scale invariance remain adequate. However, negation requires some expansion to support the new encoding values defined above.

Addendum to Equation 3.15: Partially Defined and Undefined Operational Result Negation

(M_i, C_i)	$(u,0)$	$(u,+1)$	$(+1,u)$	. . .	$(-1,u)$	$(u,-1)$	(u,u)
$\neg(M_i, C_i)$	$(u,0)$	$(u,-1)$	$(-1,u)$	. . .	$(+1,u)$	$(u,+1)$	(u,u)

Two applications of the previously defined operational template (addition and multiplication) are developed in the next section. However, an important point to consider is that these two operations are just examples of binary relations. This template allows for the definition of any binary operation. Relative to data-driven discovery, this ability to include or exclude operations demonstrates an aspect of analytical control that opens up many domains outside of materials processing for which this method was conceived.

4.2 The Operations of Addition and Multiplication

Originally, graphical experimentation was used to develop operational solutions within the set of primitive pairings. The resultant tables demonstrated promising evaluational results, however, experimental incompleteness produced several errors, and considerably more undefined regions than was desirable. The realization that monotonicity and concavity, as defined in Section 3.3, mirror basic differencing techniques, provided a considerably more complete and accurate mechanism for generating accurate operational results.

Computational differencing equates to the discrete forms of the first and second derivatives, with an underlying assumption of smoothness and differentiability. Therefore, the application of real-valued derivatives paired with the four symbols previously defined for monotonicity and concavity, allows for the reasonable computation of a resultant symbolic value, and the generation of operational tables.

4.2.1 Addition in Transform-Space. The symbolic addition of transformed signals, effectively seeks to combine two equivalence class signatures and produce a new signature, representing the class containing the transformed numerical result of the operation. The resultant signature defined by addition, must therefore, represent the set of all possible additions of the original two real-valued signals, invariant specifically to shifts and positive scales of the original two signals. Although conceptually difficult, consider that the resultant signature of any operation also repre-

sent both original signals. Consequently, the definition of a resultant equivalence class becomes a mathematical process of solving for the commonality between the two input signatures.

To solve for discrete-space addition, consider the first and second derivatives of binary addition.

Given: The First and Second Derivatives of Added Functions

$$\frac{d}{dt}(\bar{a}+\bar{b}) = \frac{d}{dt}\bar{a} + \frac{d}{dt}\bar{b}$$

$$\frac{d^2}{dt^2}(\bar{a}+\bar{b}) = \frac{d^2}{dt^2}\bar{a} + \frac{d^2}{dt^2}\bar{b}$$

The terms of the first derivative represent numeric values, however these values reveal piecewise monotonicity. Sequences of positive values indicate a monotonically increasing interval. Conversely, sequences of negative values indicate a monotonically decreasing interval. Similarly, the terms of the second derivative reveal piecewise concavity, with positive intervals indicating a convex region, and negative intervals indicating a concave region.

Some basic properties of real-valued addition allow the substitution of *DMC* transform-space symbols computationally into both terms of the first and second derivatives for addition. The first property guarantees that the addition of two positive numbers or a positive number and zero always results in a positive number. Similarly, the addition of a negative number to any other negative number or zero consistently results in a negative number. In terms of signature addition, the only unresolvable combinations add a positive and a negative number, or any symbol plus an unknown. The 'symbolic' calculations are given in Appendix A, but the results for monotonicity and concavity are separately summarized in Table 4.1.

The operation of *DMC* transform-space addition demonstrates three significant algebraic properties. First, the addition of the symbol u allows addition to remain operationally closed for monotonicity, concavity, and the combined operation of transform-space addition. Secondly,

		$\mathcal{M}$onotonicity $\mathcal{M}_i$			
	$\mathcal{M}$ +	-1	0	+1	u
$\mathcal{M}_j$	-1	-1	-1	u	u
	0	-1	0	+1	u
	+1	u	+1	+1	u
	u	u	u	u	u

		$\mathcal{C}$oncavity $\mathcal{C}_i$			
	$\mathcal{C}$ +	-1	0	+1	u
$\mathcal{C}_j$	-1	-1	-1	u	u
	0	-1	0	+1	u
	+1	u	+1	+1	u
	u	u	u	u	u

Table 4.1 **DMC Transform-Space Addition**

constant monotonicity and concavity define respective unique identities for addition. Thirdly, it can be shown that transform-space addition is associative (i.e., a+(b+c) = (a+b)+c). Appendix A exhaustively proves associativity under addition. Closure, associativity, and symbolic identity allow *DMC* transform-space addition to be classified as a 'groupoid' in terms of an abstract algebra. What can not be shown are unique symbolic inverses, which would allow this operation to be classified as a 'group' [12].

4.2.2 Multiplication in Discrete-Space. Similar to addition, multiplication seeks to combine two equivalence class signatures to produce a new resultant signature. And likewise, solving for the commonality between two input signatures is most effectively accomplished using the first and second derivatives of binary real-valued multiplication.

Given: The First and Second Derivatives of Multiplied Functions

$$\frac{d}{dt}(\bar{a} * \bar{b}) \quad = \quad \left(\frac{d}{dt}\bar{a}\right)\bar{b} + \bar{a}\left(\frac{d}{dt}\bar{b}\right)$$

$$\frac{d^2}{dt^2}(\bar{a} * \bar{b}) \quad = \quad \left(\frac{d^2}{dt^2}\bar{a}\right)\bar{b} + 2\left(\frac{d}{dt}\bar{a}\right)\left(\frac{d}{dt}\bar{b}\right) + \bar{a}\left(\frac{d^2}{dt^2}\bar{b}\right)$$

These derivatives imply that any solution to the operation of multiplication in discrete-space requires the real values, $\bar{a}$ and $\bar{b}$, that are not maintained by *DMC*. However, the property of shift invariance (Section 3.4.1) justifies the assumption that any time series can be positively shifted until

Monotonicity

$\mathcal{M}_j$ \ $\mathcal{M}_i$	-1	0	+1	u
-1	-1	-1	u	u
0	-1	0	+1	u
+1	u	+1	+1	u
u	u	u	u	u

Concavity

$\mathcal{M}_j, \mathcal{C}_j$ \ $(\mathcal{M}_i, \mathcal{C}_i)$	-1,-1	-1,0	-1,+1	0,0	+1,-1	+1,0	+1,+1	--,u
-1,-1	u	u	u	-1	-1	-1	u	u
-1,0	u	+1	+1	0	-1	-1	u	u
-1,+1	u	+1	+1	+1	u	u	u	u
0,0	-1	0	+1	0	-1	0	+1	u
+1,-1	-1	-1	u	-1	u	u	u	u
+1,0	-1	-1	u	0	u	+1	+1	u
+1,+1	u	u	u	+1	u	+1	+1	u
--,u	u	u	u	u	u	u	u	u

Table 4.2 **Discrete-Space Multiplication**

all observational values are greater than zero, without affecting the accuracy of the representation or the operation. This assumption allows the symbolic reduction of the previous derivatives, as shown below, such that monotonicity and concavity can be computed inside the discrete-space.

Given: The Reduced Derivatives of Multiplied Functions

$$\frac{d}{dt}(\mathcal{A}_i * \mathcal{B}_i) = \left(\frac{d}{dt}\mathcal{A}_i\right)(+1) + (+1)\left(\frac{d}{dt}\mathcal{B}_i\right)$$

$$\frac{d^2}{dt^2}(\mathcal{A}_i * \mathcal{B}_i) = \left(\frac{d^2}{dt^2}\mathcal{A}_i\right)(+1) + 2\left(\frac{d}{dt}\mathcal{A}_i\right)\left(\frac{d}{dt}\mathcal{B}_i\right) + (+1)\left(\frac{d^2}{dt^2}\mathcal{B}_i\right)$$

Notice however, that unlike symbolic addition, the computation of multiplicative concavity requires the inclusion of the associated monotonic terms. The complete symbolic solution for multiplication is again given in Appendix A, with the results summarized in Table 4.2.

Section 3.3.1 first commented that encoding monotonicity alone was representationally and resolutionally weaker than the pairing of monotonicity and concavity. A comparison of the monotonic operational results from Tables 4.1 and 4.2 illustrates the lack of any resolution between *DMC* transform-space addition and multiplication. The inclusion of concavity allows at least some discrimination between these two basic operations.

Referring back to abstract algebra, multiplicative monotonicity can be similarly classified as a 'groupoid' using the same reasoning as was applied to addition. The same classification can not be independently made relative to multiplicative concavity. However, considered as a pair, monotonicity and concavity demonstrate closure, associativity[2], and a unique identity. Therefore, transform-space multiplication may still be referenced as an algebraic 'groupoid'.

4.3 Strategy for Multivariate Search and Recognition

Section 3.5 illustrated the basic outline for transform-space relational discovery. This section expands that outline, first enhancing the figure of merit to include partially defined operational results for trivariate analysis, and then adding guided iterative search for multivariate analysis.

4.3.1 Trivariate Relational Discovery. Given the algebraic expansions developed in the previous two sections, trivariate analysis simplifies to a mere expansion to the original bivariate methodology illustrated in Figure 3.5. A trivariate relation maps a combination of two hypothetical or actual time series variables onto another single process variable (*e.g.* $z = c_1x + c_2y + c_3$, or $z = c_1(x * y)$). What now remains is the expansion of search and evaluation.

Trivariate search can be approached from one of two ways. The first selectively combines independent variables, possibly based on their bivariate figures of merit, for later evaluation. The second exhaustively combines all possible pairings, similar to the bivariate search. In the interest

[2]The exhaustive proof of associativity has not been included as part of Appendix A due to the extremely large number of possible combinations.

of discovering all possible relations, the later has been chosen given its general computational tractability, its breadth of search, and a lack of conclusive evidence for significant bivariate FOMs that highlight all of the terms involved in a higher order relation. As was previously stated, the two operations defined in Sections 4.2.1 and 4.2.2 allow the consideration of transform-space addition, subtraction multiplication and division.

Combined with the bivariate pairing, the resultant exhaustive trivariate search considers the following possible combinations.

A	A+B	A*B	A/B	¬A/B
¬A	¬A+¬B	A*¬B	A/¬B	¬A/¬B
A*¬A	A+¬B	¬A*B	B/A	¬B/A
A/¬A	B+¬A	¬A*¬B	B/¬A	¬B/¬A
¬A/A				

The first column is repeated for every independent time series variable. The remainder are repeated for each unique combination of two independent variables. The exhaustive size of this search-space is $O(n^2)$, with n representing the number of independent time series variables[3].

With respect to actually implementing the four transform-space operations (+, -, *, /), addition and multiplication are very straight forward. Subtraction, on the other hand, is simply the addition of a variable plus the negation of another, while division is accomplished by multiplying the relational divisor by the result for later comparison against the dividend (i.e. $\mathcal{A}_i/\mathcal{B}_i = \mathcal{C}_i$ is computed as $\mathcal{A}_i = \mathcal{B}_i * \mathcal{C}_i$).

Another important point to consider is the distribution and collection of negations within the transform-space. Relative to addition, discrete-space negation parallels its numeric-space equivalent (i.e. $\neg(\bar{\mathcal{A}} + \overline{\mathcal{B}}) = \neg\bar{\mathcal{A}} + \neg\overline{\mathcal{B}} = \neg\bar{\mathcal{A}} - \overline{\mathcal{B}}$) However, relative to multiplication, discrete-space negation

[3]The actual dimension of the search-space is $5n + 16\frac{n!}{(n-2)!*2!} = 8n^2 - 3n$

is slightly different than its numeric-space equivalent. Negatively scaling the product of two time series variables in numeric-space effectively inverts the result of the multiplication. In discrete-space, this same operation requires taking the product of the negative signatures of both variables (i.e. $-1(\bar{a} * \bar{b}) = \neg(\bar{\mathcal{A}} * \bar{\mathcal{B}}) = (\neg\bar{\mathcal{A}}) * (\neg\bar{\mathcal{B}})$). This difference becomes apparent when considering the sorted results of speculated relations.

In terms of evaluation, equation 3.16 defined the original figure of merit over monotonicity and concavity before u was added as a symbol. The inclusion of partially defined results divides an expanded FOM calculation into two parts. Partially defined regions allow for a valid range of the seven original primitives, and therefore require specific range versus equality checking. Therefore, the FOM can now be expressed as the sum of well-defined equality plus valid partially defined ranged-equality.

Definition: Uncompressed Multivariate Figure of Merit

$$FOM(\bar{\mathcal{A}}, \bar{\mathcal{B}}) = \frac{\sum_{i=2}^{n-1} \chi_{WD}(\mathcal{A}_i, \mathcal{B}_i)(t_i - t_{i-1}) + \sum_{i=2}^{n-1} \chi_{PD}(\mathcal{A}_i, \mathcal{B}_i)(t_i - t_{i-1})}{2(t_{n-1} - t_1)} \tag{4.3}$$

$$\text{where} \quad \chi_{WD}(\mathcal{A}_i, \mathcal{B}_i) = \begin{cases} 1 & \text{if } \mathcal{A}_i = \mathcal{B}_i \\ 0 & \text{otherwise} \end{cases}$$

$$\text{and} \quad \chi_{PD}(\mathcal{A}_i, \mathcal{B}_i) = \begin{cases} 1 & \text{if } \mathcal{A}_i.M = \mathcal{B}_i.M \ \wedge \ (\mathcal{A}_i.C \vee \mathcal{B}_i.C) = u \\ 1 & \text{if } \mathcal{A}_i.C = \mathcal{B}_i.C \ \wedge \ (\mathcal{A}_i.M \vee \mathcal{B}_i.M) = u \\ 0 & \text{otherwise} \end{cases}$$

Having defined the necessary expansions to bivariate search and recognition, an example of the resultant method for trivariate relational discovery can be considered. Figure 4.2 illustrates an example of additive trivariate relational discovery. Illustrationally, uncompressed signatures

visually maintain the periodicity of the original waveform, and are therefore preferable in terms of display.

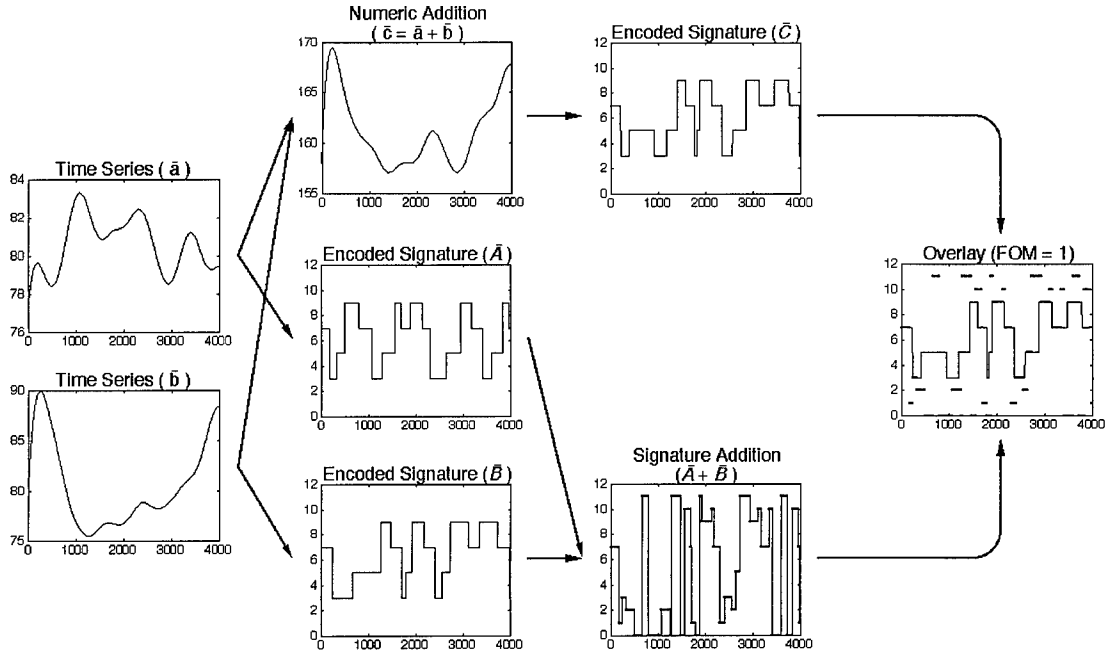


Figure 4.2 **Symbolic DMC Signature Addition.** Three real-valued time series ($\bar{a}$, $\bar{b}$, and $\bar{c}$ which equals the simple addition of the first two) are considered as input for relational analysis. All three are subsequently encoded using the first two component of the *DMC* transform. Finally, the evaluation of the symbolic, signature addition of $\bar{A} + \bar{B}$ produces a figure of merit equal to one, indicating a perfectly correlated candidate relation.

To highlight the relational evaluation in the previous example, Figure 4.3 enlarges the overlay of the symbolic operational signature over the encoded mathematical result from Figure 4.2. Plainly, well defined operational results overlap within the range of the seven original primitives, while partial results fall above and below the encoded mathematical variable. The matching of negative partially defined monotonicity and convex (positive) partially defined concavity have been outlined to illustrate both the value of partial definitions, and the actual range checking that is required in

terms of the FOM. Notice also that in terms of this example, approximately 25% of the resultant signature remains completely undefined. However, this percentage varies dramatically depending on the encoded variables and the operation performed.

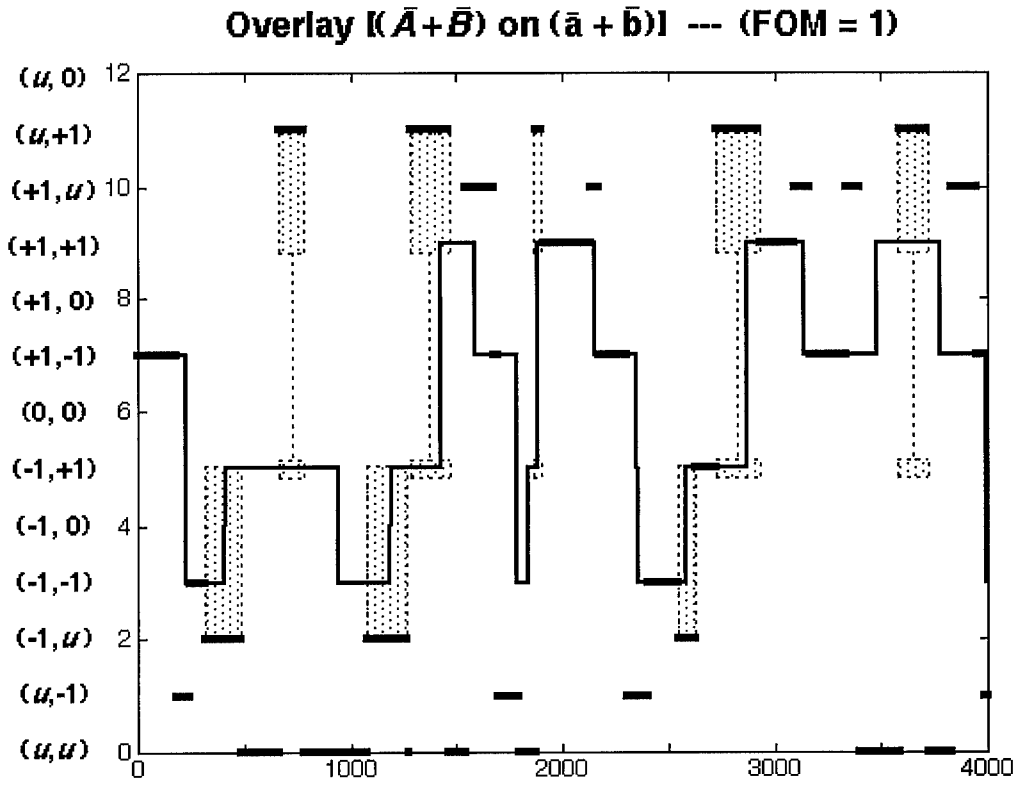


Figure 4.3 **Overlay of a Symbolic Partial Signature on an Encoded Mathematical Result.**

As one final example before considering a further expansion, Figure 4.4 illustrates coefficient invariance relative to operational results. The addition of a scalar coefficient to the example presented in Figure 4.2 can greatly affect the resultant waveform, and consequently, the resultant encoding. However, overlay of the symbolic operational signature produces equivalent results, effectively isolating scale-dependent intervals inside of partially or undefined regions of the signature.

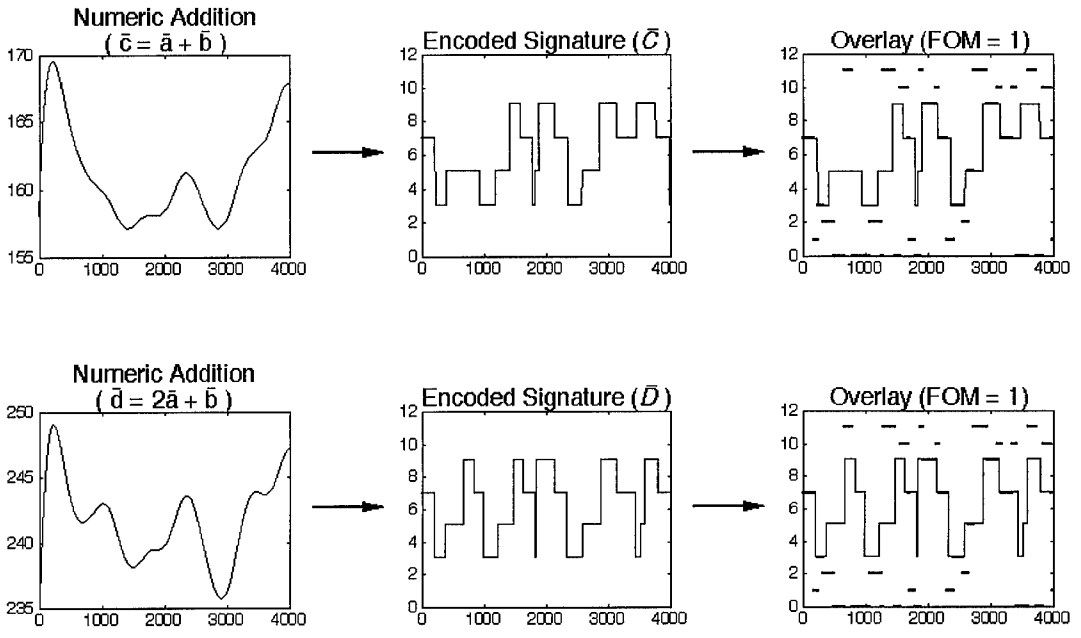


Figure 4.4 **Demostration of Coefficient Signature Additon.** This figure is a continuation of Figure 4.2.

4.3.2 Expansion to Multivariate Relational Discovery. The previous section outlines the basic application of *DMC* for bi/trivariate relational discovery. This section proposes extending that method, based upon experimental results presented in Chapter V, into even higher order analysis. Because exhaustive multivariate relational analysis would, of course, become computationally intractable, higher order analysis is carried out via the injection of highly correlated lower order signatures into successive iterations of combination and evaluation.

The functional premise of this iterative approach forwards the operational signatures of a lower order relational terms for further combination. For example, if the relation to be discovered is $\bar{\mathcal{A}}+\overline{\mathcal{B}}+\overline{\mathcal{C}} = \overline{\mathcal{X}}$, then forwarding either $\bar{\mathcal{A}}+\overline{\mathcal{B}}$, $\bar{\mathcal{A}}+\overline{\mathcal{C}}$, or $\overline{\mathcal{B}}+\overline{\mathcal{C}}$ allows for the subsequent combination of the remaining term, and the potential discovery of $\mathcal{X}$. Similar to the BACON system (reference

Section 2.3), forwarded signatures are simply considered additional variables for combination and evaluation.

In terms of relation evaluation, combinational progression from forwarded results towards higher order relations is guaranteed to produce a FOM of equal or greater value than the forwarded lower order term, relative to any associate operation. The inherent hazard is that each combination will compound the previous loss in resolution due to the increasing number of unsolvable intervals.

Figure 4.5 expands the previous methodology (Figure 3.5) to support iteration, and signature forwarding. An additional component representing algebraic knowledge has been included to prevent unproductive cycling between successive iterations of the search. However, some additional efficiency is possible by integrating this knowledge to prune prior to generating combination which undo previous combinations.

This model for multivariate analysis requires the addition of two configurational parameters to the system. The first defined the number of operational signatures to be forwarded between successive iterations. The second simply defines the number of iterations to be processed. Hard-coding the number of iterations is a current limitation of this method. Ideally, the system should either continue searching as long as time permits, or should have some way of recognizing when to stop iterating.

Figure 4.5 completes the *DMC* transform-space methodology for multivariate relational discovery. The next chapter present the initial experimental results of this method.

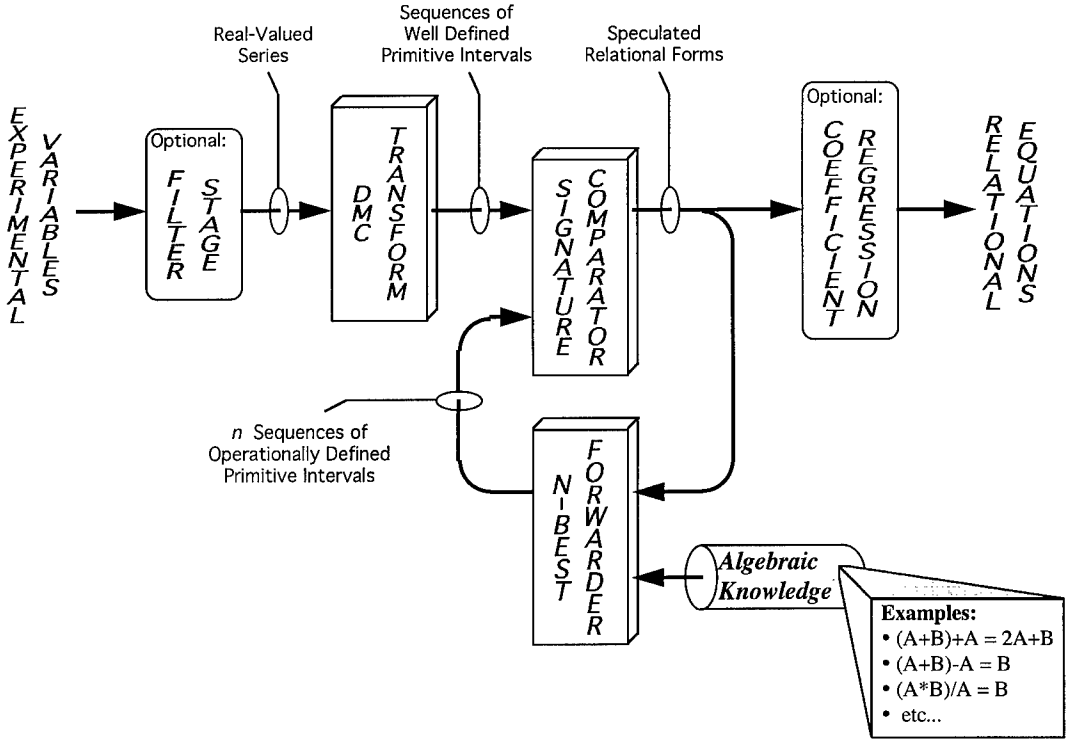


Figure 4.5 **Multivariate Relational Discovery.**

V. *Experimental Results*

This chapter documents the initial testing of this method for multivariate relational discovery, as illustrated in Figure 4.5. The first section explains the experimental setup used for testing. Then, Section 5.2 annotates the results of several artificial bivariate, trivariate, and multivariate tests.

5.1 Test Setup

As previously stated, the basic methodology for multivariate analysis has been applied to a number of artificial experiments. Prototyping, data generation and testing were exclusively conducted in MATLAB[1] for the Macintosh, version 4.2c.1, on a Power Macintosh 7100/80. The average execution time, without signature compression, for five iterations, given nine initial time series, and forwarding five candidates per iteration, was six hours.

The methodology presented in Section 4.3.2 was implemented with one significant regrettable exception. Instead of forwarding the operational signature as discussed in Chapter IV, the *best-n* candidates were numerically computed with normalized real-values, encoded, and then returned to the signature comparator. This decision was originally based on early operational tables and a focus on the large percentages of undefined regions generated in operational results. Normalization attempted to counter the effects of very large values overriding the relational contributions of very small values, but in essence, this decision arbitrarily fixed time-series scale factors and relational coefficients.

The effects of this decision degrade multivariate relational discovery, and are highlighted in Section 5.2. Subsequent to this decision, the operational resolution for addition and multiplication was significantly improved by the application of the first and second derivatives, as developed in Chapter IV. This improvement, coupled with some additional consideration given in Chapter VI,

[1]MATLAB is a registered trademark of The Math Works, Inc.

should more strongly support the application of this method to multivariate analysis as presented in Section 4.3.2.

In term of the artificial time series data, 9 sequences of 10,000 normally distributed, random observations were generated inside of MATLAB. The absolute value was then taken to combine the random variation above the original mean. Lastly, each series was filtered with a 3^{rd} order low-pass digital Butterworth filter.

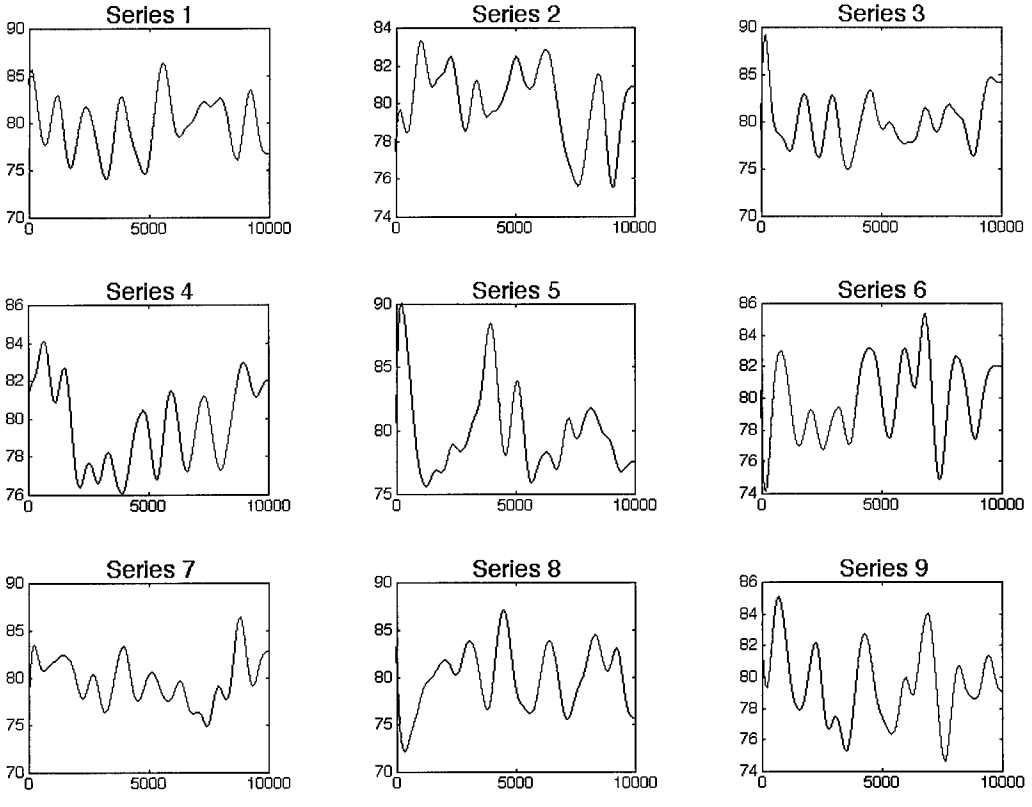


Figure 5.1 **Randomly Generated Experimental Time Series.**

The resultant artificial series are shown in Figure 5.1. The time series generation process attempts to eliminate any accidental dependencies between the experimental variables, so that only induced experimental relations would be evaluated during testing.

All nine of these experimental time series were provided as input to the prototype system. One additional hard-coded mathematical combination of those nine was then defined for 'relational discovery'. Case specific definition and results of this testing are provided in the next section.

5.2 Annotated Results

Section 3.5 presented the initial methodology for bivariate relational discovery, and expressed that the space of bivariate relations would be fully traversed in the first iteration of multivariate analysis. The following table documents seven experiments with artificial bivariate relations.

Artificial Bivariate Relational Testing

input series	scale factor	forward/ iterations	final sorted position	total relations considered	other of $\geq$ FOM by iteration 1	2	3	4	5
3	1	5/5	1	6613	48	98	156	226	270
5	-1	5/5	1	6613	48	98	156	225	275
3	-1/3	5/3	1	2813	48	98	156		
7	1/15	5/3	1	2813	48	98	156		
9	-256	5/4	1	4513	48	98	156	226	
2	6001	5/4	1	4513	48	98	156	226	
4	-1/6	5/5	1	6613	48	98	156	226	275

The first column of the preceding table indicates which of the previously illustrated nine time series is to be discovered, while the second column represent a specific scalar multiple applied to that series. The third column documents the configurational parameters for the system (i.e. the number of relation the forward during each iteration, and the number of iterations to process) used

during each test. The next columns identify the line number containing the correct hard-coded relation inside the sorted list of processed relations, followed by the total number of relations that were evaluated during the entire experiment. The last five columns indicated the growth in number of other processed relations with figures of merit greater than or equal to the correct experimental relation by processing iteration.

The first table documents *DMC*'s remarkable ability to discover noise-free bivariate relations, however, it also indicates that the system also speculates an increasing number of spurious relations as the number of iterations increases. This effect represents one current resolutional side-effect of the two discrete-space operations. Currently, operational combinations of any variable with a correctly identified bivariate term, produces a resultant signature of equal or potentially greater FOM. These operational combination justify the regular pattern of growth indicated in the last five columns of the bivariate test results.

It is hoped that some of the potential resolutional enhancements discussed in the next chapter will correct this side-effect. Currently, this side effect seems to diminish in higher order relations.

The next two tables similarly present the results for additive and multiplicative trivariate relational discovery. In all eighteen of the following tests, five iterations evaluate 6,613 candidate relations. The total number of well and partially defined operational results have been included for each experimental relation, to illustrate the current resolutional decay after one operation.

As expected, trivariate relations discovery demonstrates equally remarkable performance. Additionally, the previous side-effect appear substantially diminished in all but the fourth additive case. However, in that fourth test, a very large scale was applied to one time series variable, while the other was divided by two. The large difference in scaling actually hides the second variable such that the first matched independently as a simple bivariate relation. In this one case the position of the correct relation was coincidentally linked to the side-effect previously discussed.

Artificial (Additive) Trivariate Relational Testing

input relation	scale factors		forward/ iterations	final sorted position	total well def (% correct)	total partially (% correct)	other of $\geq$ FOM
6+9	1	1	5/5	1	5761 (100%)	4101 (100%)	34
7-4	1	1	5/5	1	2851 (100)	4640 (100)	11
3-5	27	14	5/5	1	3671 (100)	4555 (100)	15
1+4	14000	1/2	5/5	4	1129 (100)	4629 (100)	270
7-6	1/7	1/18	5/5	1	4703 (100)	4381 (100)	11
9-2	164	17	5/5	1	2558 (100)	5534 (100)	17
2+8	1	2470	5/9	1	3217 (100)	4522 (100)	197

Artificial (Multiplicative) Trivariate Relational Testing

input relation	scale factors		forward/ iterations	final sorted position	total well def (% correct)	total partially (% correct)	other of $\geq$ FOM
6*3	1	1	5/5	1	1932 (100%)	5157 (100%)	26
7*1	1	1	5/5	1	1243 (100)	4608 (100)	17
8*9	1/17	7	5/5	2	1925 (100)	5476 (100)	12
6*2	601	38	5/5	2	1034 (100)	6212 (100)	17
4/9	1/2	13	5/5	3	237 (100)	2164 (100)	12
5/2	1024	6	5/5	1	906 (100)	3400 (100)	8
1/7	1	1	5/9	3	355 (100)	3556 (100)	18
8/6	70	1/21	5/5	1	276 (100)	4415 (100)	2
3*1	1	-1	5/5	41	756 (99.21)	4949 (100)	40
5*2	1	-1	5/5	7	2338 (99.96)	5355 (100)	7
9*4	-1	-1	5/9	2	722 (100)	4919 (100)	8

Also of note, the last three multiplicative trivariate tests demonstrated the negative-numeric to negated-signature relations that was introduced in Section 4.3.1. The resultant notation for the three discovered relations was $(\neg 1 * \neg 3)$, $(\neg 2 * \neg 5)$, and $(4 * 9)$ respectively.

The last three tables document several tests attempting higher order relational discovery. Recognizing the previously stated deficiency in forwarding operational results, these tables focus on highlighting the emergent positions and relative figures of merit of potentially traceable lower order terms. Notationally, the positions of all lower order terms are relative to the 613 combinations of the first iteration. 'Failure' generally indicates that the search had not yet forwarded necessary lower order terms.

In terms of the additive and multiplicative tests, forwarding any one of the lower order terms allows the evaluation of the four-variable relations. Of particular interest, is the fourth additive test, which was at least successfully processed the correct multivariate relation. In this case, the multivariate figure of merit actually decreased from the forward lower order term. This decrease is directly related to the arbitrary fixing of time-series scale factors and relational coefficients. Such a decrease is mirrored in the first two multiplicative four-variable tests.

Artificial (Additive) Four-Variable Relational Testing

input relation	scale factors	iter found	final pos	% FOM	other of $\geq$ FOM	1^{st}, 2^{nd}, 3^{rd} lower-order (2^{nd} iter pos) % FOM		
3+5+1	1 1 1	2	7	99.81%	7	(2) 92.62%	(26) 77.63%	(70) 66.44%
4+2+7	1 1 1	2	36	98.39	35	(2) 92.82	(8) 85.73	(45) 70.0
2-9-4	11 1 301	failed	n/a	n/a	n/a	(38) 97.55	(39) 97.51	(410) 28.78
6-8+1	1/16 7 23	2	770	88.26	769	(1) 100	(56) 83.02	(263) 40.63
6-2-9	1 1 1	failed	n/a	n/a	n/a	(11) 71.66	(26) 69.34	(33) 65.05

Artificial (Multiplicative) Four-Variable Relational Testing

input relation	scale factor	iter found	final pos	% FOM	other of ≥ FOM	1^{st}, 2^{nd}, 3^{rd} lower-order (2^{nd} iter pos) % FOM		
4*5*8	1	2	498	76.81%	498	(2) 87.46%	(32) 65.13%	(38) 63.04%
1*2*3	7	2	844	80.0	843	(5) 94.69	(54) 70.85	(68) 65.49
7*9/6	1	2	67	94.21	66	(3) 87.66	(143) 55.39	(280) 40.03
2/6/5	3	failed	n/a	n/a	n/a	(81) 58.86	(161) 47.95	(579) 10.15
8*7/1	1	failed	n/a	n/a	n/a	(6) 90.53	(31) 83.19	(69) 70.55

Artificial (Mixed) Four-Variable Relational Testing

input relation	scale factors	iter found	1^{st}, 2^{nd}, 3^{rd} lower-order (2^{nd} iter pos) % FOM		
(7+1)/4	1	failed	(14) 84.63%	(18) 83.30%	(77) 67.30%
3*(5+1)	1	failed	(4) 96.33	(11) 85.13	(168) 52.74
4/(5-9)	1	failed	(593) 0.96	***	***

VI. For Future Consideration

The previous chapter documented a number of implementational deficiencies within the current system. This chapter documents several enhancements that are currently considered for future implementation.

6.1 Continuing Discrete-Space Search

Section 4.3 introduces the desire to iterate exclusively inside of *DMC*'s discrete-space, prefacing experiment support in Chapter V. As experimentally demonstrated, numerically computing intermediate binary combinations for iterational forwarding arbitrarily fixes scale factors and relational coefficients, consequently biasing forwarded results and obfuscating the multivariate signatures that the method is attempting to pursue. Operations in discrete-space require no such assumptions of scale, and consequently do not induce a bias in terms of successive operations.

Any ability to continue 'operating' inside of the discrete-space is currently limited by the resolution of operational results, and the concession allowing partially defined and undefined regions. If, for example, each operation over two completely defined series produced only a 50% well defined operational result, then accurate evaluation becomes proportional to the number of variables potentially involved.

Therefore, resolution management is the key to this operational shortcoming. The next section presents three enhancements that address improving *DMC* resolution in terms of this analytical method. However, any resolutional enhancement must be carefully evaluated to avoid potentially 'resolving' away the incremental multivariate signatures used to guide higher order search.

6.2 Addressing Better Resolution

Chapter IV introduced the rationale for and some of the associated problems with less than"well defined" operational results in terms of higher dimensional analysis. Unfortunately,

ambiguous regions hinder accurate resolution both inside the iterative search, and also in terms of evaluating the results produced from this method. The necessity for efficient evaluation was stressed in Chapter I, even above that of efficient search. This section presents three potential enhancements considered for future implementation.

Improving the Figure of Merit. Ambiguous regions are currently overlooked in the figure of merit equation used for evaluation. Partially defined matches are weighted equally to complete matches, and undefined resultants are not considered at all. In the author's opinion, any penalty assessed solely on operationally undefined or partially defined regions would adversely affects this method in terms of those operations. Such a penalty would imply that some pairings are more significant than others, which is not the case. On the other-hand, alternate figures of merit, such as separating monotonic correlations from that of concavity might demonstrate that certain derivatives are more important relative to relational discovery than others. Another possibility would evaluate incrementally along the derivative orders. Such an incremental evaluation would compute the monotonic correlation separately, and then consider monotonicity and concavity jointly, and so on. These alternatives represent just two possibilities that may improve evaluation within current operational resolutions. Additionally, these two alternatives foreshadow the next potential enhancement.

Adding Higher Order Derivatives. The *DMC* representation as described in Chapters III and IV incorporates aspects of the first and then the second order derivatives, consequent to their visual significance. Although higher order derivatives potentially lose simple visual significance, successive orders may hold yet undiscovered relational significance. In such a case, consideration of the additional complexity must be weighed against the potential resolutional improvement. The addition of such terms might substantially increase the number of partially defined and undefined operational pairings, as well as decrease processing speed. However, higher order terms may also

increase the representational and more importantly, operational resolutions such that multivariate analysis exclusively in discrete-space becomes realistically possible.

Inserting Sequences into Undefined Regions. As explained in Chapter IV, the operational combination of two temporal regions is not necessarily guaranteed to produce only one resultant region. This fact underpins the currently undefined and partially defined regions hindering continued operational search inside discrete-space. What has yet to be addressed are potential limits on the number of valid sequences generated in such regions. In terms of any partially defined region, there is only one degree of freedom. Intuition suggests that many such partially defined regions will change at most once, with respect to that degree of freedom, given smooth waveforms. Is it then equally valid in the case of some undefined resultants, to suggest that within those regions each of the two degrees of freedom will change at most once? In either case, the temporal instant of these inflections would not be computable in discrete-space, but such insight might allow the number of potential sequences within a region to be quantified for conditional evaluation.

For example, the addition of an (increasing, concave) segment with a (decreasing, concave) segment results in a concave segment with undefined monotonicity (see. Table 4.2.1). The first term's rate of change is increasing, while the second term's rate of change is decreasing. Therefore, it is reasonable to assume that if the rate of change of the second initially exceeded the first, but then the first term's rate overtakes the second, then the first will continue to dominate from that instant. Reasonably, the set of possible sequences within the region of overlap given this pairing and operation could be reduced to [(dec,conc) ; (dec,conc),(inc,conc) ; (inc,conc)]. This reduction would allow the actual region temporally equivalent to the region of overlap to be conditionally evaluated against the three possible resultants. Then, matching cases would lend additional support to the relation being evaluated, while non-matchable cases might tend to invalidate the relation.

6.3 Residual Analysis

Often, it is not feasible or cost effective to measure all of the desired variables for any given process. Additionally, unknown or unrecognized variables may exist that have yet to be considered. Such 'unknowns' often represent critical pieces of information, necessary for understanding the dynamics of a process. Therefore, any classification even of the form of an unmeasured or unknown variable may be of enormous value.

This method has demonstrated the ability to discovery relations between measured inputs and outputs. It would seem possible, however, given the combinational algebra described in Chapter IV, to at least partially compute an unmeasured quantity at least in form. In such cases, simple linear components, or possibly speculated forms that fit into multiple relations might provide cues to the existence of other variables. Conceivably, any such technique would be limited to speculating a single form representing a possible set of unknown variables.

Additionally, if regression is applied to solve for the coefficients of a relational form discovered using *DMC*, then the residual of that fit may contain interesting information. In the case of the PLD example shown in Figure 3.1, fitting the filtered laser energy signal to the spectral measurement reveals a simple linear component, possibly representing decay. Patternistic residual analysis is an additional major research problem [20], however, this discovery method may allow for a simple solution. Time allowing, residual signals could be injected into a second pass of this method, allowing residual relational discovery to proceed simply from the larger set.

6.4 Neural Considerations

Neural networks have demonstrated remarkable potential for learning and time series prediction. Although currently unprecedented, neural architectures exist that may be adaptable to this more explanatory time series problem. Combining relevant theories for the extraction of coherent rules from the distributed information contained in a network's relative weights, with one or more

appropriately structured networks, might produce a relational discovery system of equal or greater efficiency than the previous method. Additionally, it is conceivable that a neural approach that processed the transformed information presented in this research, could more efficiently search the problem space.

6.5 Beyond Discovery

Outside of this method for relational discovery, the techniques developed in Chapters III and IV could be applied to many other processing areas. Trivially, these techniques could be combined with a library of template patterns such as sine, square, etc., for signal identification and periodic characterization. Along those same lines, signal addition or multiplication by similarly templated noise could then be matched against actual data conceivably to characterize signal noise. However, the second application is not so trivial.

The major difficulty for symptom-based fault detection is knowledge acquisition [8]. Symptom-model-based approaches to fault detection combine heuristic symptoms with system inputs to monitor and recognize faults within a process. Currently, *DMC* is designed strictly for post-processing. But, assuming sufficient improvement to support real-time operation, this method could autonomously generate heuristic relations through simple monitoring. These relations could then be tracked, and if violated, simply raise the potential faulting conditions. However, significant testing and considerable improvements are necessary before any such application could be realized.

VII. Conclusions

Various authors have downplayed the potential contributions of exclusively data-driven approaches to relational discovery. It has been suggested that purely data-driven discovery is often impossible, and in any case much more difficult than is often assumed [4]. Another argument suggests that this type of discovery does not entail most of the activities involved in empirical research, such as experimental design, or hypothesis testing and theory revision [3]. Granting that these discovery methods will not replace a research scientist, hopefully, the *DMC* transform and its associated method for relational discovery have restated the conclusions originally drawn from BACON, that automated data-driven discovery is both plausible and computationally tractable.

This thesis presents a new approach to signal analysis and relational process discovery. Chapters III and IV develop several autonomous mechanisms which implement Gerwin's four aspects for extracting relations from data (i.e. pattern perception, classification, class specific resolution, and recycling, if necessary). This method also extends beyond simple linear or bivariate relations to address the larger issue of multivariate linear and non-linear relational discovery from primarily non-linear 'real' data.

Algorithmic *DMC* encoding and compression of time series signals offers substantial representational contributions to data-driven relational discovery. *DMC*'s representational properties of shift and scale invariance eliminate two infinite degrees of freedom. Likewise, the reduction of continuous time series values to 13 discrete primitives greatly simplifies comparitive evaluation.

The foremost contribution, however, is the ability to 'algebraically' and associatively combine discrete-space signatures to produce new signatures representative of all the possible combinations of the orignal signals via specific operations. This ability combined with appearantly traceable lower order signatures provides substantial potential for computationally tractable, autonomous, multivariate relational discovery.

The future considerations presented in Chapter VI represent significant, achievable improvements to the foundations demonstrated by this research. These improvement should also serve to correct the problems noted during experimental testing, and it is the intent of this author to continue developing *DMC*, specifically attempting to produce a low speed real-time system capable of actual process monitoring and fault detection.

The basic premise for operationally combining compressed data signatures offers a significant contribution to artificial discovery, while the fundamental idea may be applicable to other areas. Combination of this technique with others, such as Schaffer's $E*$ algorithm, may demonstrate a much greater resolutional ability to discover and model experimental processes. Given the ever increasing volume of collected data, techniques such as *DMC* will be increasingly called upon to efficiently reduce 'real' data down to accurate relations.

Bibliography

1. Abrams, Frances L. *Process Discovery: Automated Process Development for the Control of Polymer Curing*. PhD dissertation, University of Dayton, December 1995.

2. Bassetti, D., et al. *Estimates for Material Properties: The Method of Multiple Correlation*. Technical Report CUED/C-EDC/TR33, Engineering Design Centre, University of Cambridge, January 1996.

3. Borowski, E. J. and J. M. Borwein. *The Harper Collins Dictionary of Mathematics*. Harper Perennial, 1991.

4. Chatfield, Christopher. *The Analysis of Time Series, An Introduction*. Chapman and Hall Ltd., 1984.

5. Cheung, Jeff and Jim Horwitz. "Pulsed Laser Deposition History and Laser-Target Interactions," *Materials Research Society Bulletin*, 30–36 (February 1992).

6. Council, National Research. *Mathematical Challenges from Theoretical/Computational Chemistry*. National Academy Press, 1995.

7. Devaney, Judith E. "A Machine Learning and Equation Signature Approach to Equation Discovery." *Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium*. 111–115. 1995.

8. Frank, Paul M. and Birgit Köppen-Seliger. "New Developments Using AI in Fault Diagnosis." *1995 IFAC/IMACS International Workshop on Artificial Intelligence in Real-Time Control*. 1–12. 1995.

9. Garrett, Patrick H. *Computer Interface Engineering for Real-Time Systems*. Prentice Hall, Inc., 1987.

10. Gerwin, Donald. "Information Processing, Data Inferencing, and Scientific Generalization," *Behavioral Science*, *19*:314–325 (1974).

11. Kocabas, Sakir. "Computational Models of Scientific Discovery," *The Knowledge Engineering Review*, *6*(4):259–305 (1991).

12. Laatsch, Richard. *Basic Algebraic Systems An Introduction to Abstract Algebra*. McGraw-Hill, 1968.

13. Langley, Pat, et al. *Scientific Discovery Computational Explorations of the Creative Processes*. MIT Press, 1987.

14. Langley, Pat and Jan M. Zytkow. "Data-Driven Approaches to Empirical Discovery," *Artificial Intelligence*, *40*:283–312 (1989).

15. Mandel, John. *The Statistical Analysis of Experimental Data*. Interscience Publishers, 1964.

16. Marvasti, Farokh A. *A Unified Approach to Zero-Crossings and Nonuniform Sampling of Single and Multidimensional Signals and Systems*. Illinois Institute of Technology, 1987.

17. Milosavljević, Aleksandar. "Minimal Length Encoding Methods in Molecular Biology." *Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium*. 9–11. 1995.

18. Nordhausen, Bernd and Pat Langley. "An Integrated Framework for Empirical Discovery," *Machine Learning*, *12*:17–47 (1993).

19. Rao, R. Bharat and Stephen C-Y. Lu. "A Knowledge-Based Equation Discovery System for Engineering Domains," *IEEE Expert*, 37–42 (August 1993).

20. Schaffer, Cullen. "Bivariate Scientific Function Finding in a Sampled, Real-Data Testbed," *Machine Learning*, *12*:167–183 (August 1993).

21. Simon, Herbert A. "What is a Systematic Method of Scientific Discovery?." *Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium*. 1–2. 1995.

22. Valdez-Perez, Raul E. "Generic Tasks of Scientific Discovery." *Systematic Methods of Scientific Discovery: Papers from the 1995 Spring Symposium*. 23–28. 1995.

23. Weigend, Andreas S. and Neil A. Gershenfeld. *Time Series Prediction, Forcasting the Future and Understanding the Past*. Addison-Wesley Publishing Company, 1995.

Appendix A. DMC Transform-Space Operational Solutions

A.1 Transform-Space Addition

The following table 'symbolically' computes the operational results for addition. Because the first and second derivatives for addition, as shown below, are identical, the resultant tables for monotonicity and concavity are also identical. Therefore, only the monotonic half of the computations are given.

$$\frac{d}{dt}(\mathcal{A}+\mathcal{B}) \quad = \quad \frac{d}{dt}\mathcal{A} + \frac{d}{dt}\mathcal{B} \quad \equiv \quad \overline{\mathcal{M}}_A + \overline{\mathcal{M}}_B$$

$$\frac{d^2}{dt^2}(\mathcal{A}+\mathcal{B}) \quad = \quad \frac{d^2}{dt^2}\mathcal{A} + \frac{d^2}{dt^2}\mathcal{B} \quad \equiv \quad \overline{\mathcal{C}}_A + \overline{\mathcal{C}}_B$$

Symbolic Computation of Monotonicity

(and Concavity) Under Addition

$\mathcal{M}_i$	$\mathcal{M}_j$	$\mathcal{M}_i + \mathcal{M}_j$	Result
+1	+1	(+1) + (+1)	+1
+1	0	(+1) + (0)	+1
+1	-1	(+1) + (-1)	u
+1	u	(+1) + (u)	u
0	+1	(0) + (+1)	+1
0	0	(0) + (0)	0
0	-1	(0) + (-1)	-1
0	u	(0) + (u)	u

$\mathcal{M}_i$	$\mathcal{M}_j$	$\mathcal{M}_i + \mathcal{M}_j$	Result
-1	+1	(-1) + (+1)	u
-1	0	(-1) + (0)	-1
-1	-1	(-1) + (-1)	-1
-1	u	(-1) + (u)	u
u	+1	(u) + (+1)	u
u	0	(u) + (0)	u
u	-1	(u) + (-1)	u
u	u	(u) + (u)	u

Secondly, the following table exhaustively proves additive associativity. Again, only one table is given to demonstrate associativity for both monotonicity and concavity.

Additive Associativity Proof

$\mathcal{M}_i$	$\mathcal{M}_j$	$\mathcal{M}_k$	$\mathcal{M}_{i+j}$	$\mathcal{M}_{j+k}$	$\mathcal{M}_{i+k}$	$\mathcal{M}_{(i+j)+k}$	$\mathcal{M}_{i+(j+k)}$	$\mathcal{M}_{(i+k)+j}$
+1	+1	+1	+1	+1	+1	+1	+1	+1
0	0	0	0	0	0	0	0	0
-1	-1	-1	-1	-1	-1	-1	-1	-1
u	u	u	u	u	u	u	u	u
+1	+1	0	+1	+1	+1	+1	+1	+1
+1	+1	-1	+1	u	u	u	u	u
+1	+1	u	+1	u	u	u	u	u
0	0	+1	0	+1	+1	+1	+1	+1
0	0	-1	0	-1	-1	-1	-1	-1
0	0	u	0	u	u	u	u	u
-1	-1	+1	-1	u	u	u	u	u
-1	-1	0	-1	-1	-1	-1	-1	-1
-1	-1	u	-1	u	u	u	u	u
u	u	+1	u	u	u	u	u	u
u	u	0	u	u	u	u	u	u
u	u	-1	u	u	u	u	u	u
+1	0	-1	+1	u	-1	u	u	u
+1	0	u	+1	u	u	u	u	u
+1	-1	u	u	u	u	u	u	u
0	-1	u	-1	u	u	u	u	u

A.2 Transform-Space Multiplication

The following tables 'symbolically' compute the operational results for multiplication. The reduced first and second derivatives, as justified in Section 4.2.2, are shown below. In this case the properties of numeric addition used in the previous operations must be combined with some numeric properties of multiplication. The two important properties are: the multiplication of any positive and negative number always results in a negative number, and secondly, the multiplication of any two negative numbers always results in a positive number. The computations for monotonicity and concavity are given in turn.

$$\frac{d}{dt}(\mathcal{A} * \mathcal{B}) \quad = \quad \left(\frac{d}{dt}\mathcal{A}\right)(+1) \; + \; (+1)\left(\frac{d}{dt}\mathcal{B}\right)$$

$$\frac{d^2}{dt^2}(\mathcal{A} * \mathcal{B}) \quad = \quad \left(\frac{d^2}{dt^2}\mathcal{A}\right)(+1) \; + \; 2\left(\frac{d}{dt}\mathcal{A}\right)\left(\frac{d}{dt}\mathcal{B}\right) \; + \; (+1)\left(\frac{d^2}{dt^2}\mathcal{B}\right)$$

Symbolic Computation of Monotonicity Under Multiplication

$\mathcal{M}_i$	$\mathcal{M}_j$	$(\mathcal{M}_i)(+1) + (+1)(\mathcal{M}_j)$	Result
+1	+1	(+1)(+1) + (+1)(+1) = (+1) + (+1)	+1
+1	0	(+1)(+1) + (+1)(0) = (+1) + (0)	+1
+1	-1	(+1)(+1) + (+1)(-1) = (+1) + (-1)	u
+1	u	(+1)(+1) + (+1)(u) = (+1) + (u)	u
0	+1	(0)(+1) + (+1)(+1) = (0) + (+1)	+1
0	0	(0)(+1) + (+1)(0) = (0) + (0)	0
0	-1	(0)(+1) + (+1)(-1) = (0) + (-1)	-1
0	u	(0)(+1) + (+1)(u) = (0) + (u)	u

$\mathcal{M}_i$	$\mathcal{M}_j$	$(\mathcal{M}_i)(+1) + (+1)(\mathcal{M}_j)$	Result
-1	+1	(-1)(+1) + (+1)(+1) = (-1) + (+1)	u
-1	0	(-1)(+1) + (+1)(0) = (-1) + (0)	-1
-1	-1	(-1)(+1) + (+1)(-1) = (-1) + (-1)	-1
-1	u	(-1)(+1) + (+1)(u) = (-1) + (u)	u
u	+1	(u)(+1) + (+1)(+1) = (u) + (+1)	u
u	0	(u)(+1) + (+1)(0) = (u) + (0)	u
u	-1	(u)(+1) + (+1)(-1) = (u) + (-1)	u
u	u	(u)(+1) + (+1)(u) = (u) + (u)	u

Symbolic Computation of Concavity Under Multiplication

$(\mathcal{M}_i, \mathcal{C}_i)$	$(\mathcal{M}_j, \mathcal{C}_j)$	$(\mathcal{C}_i)(+1) + 2(\mathcal{M}_i)(\mathcal{M}_j) + (+1)(\mathcal{C}_j)$	Result
(-1,-1)	(-1,-1)	(-1)(+1) + 2(-1)(-1) + (+1)(-1) = (-1) + (+1) + (-1)	u
(-1,-1)	(-1, 0)	(-1)(+1) + 2(-1)(-1) + (+1)(0) = (-1) + (+1) + (0)	u
(-1,-1)	(-1,+1)	(-1)(+1) + 2(-1)(-1) + (+1)(+1) = (-1) + (+1) + (+1)	u
(-1,-1)	(0, 0)	(-1)(+1) + 2(-1)(0) + (+1)(0) = (-1) + (0) + (0)	-1
(-1,-1)	(+1,-1)	(-1)(+1) + 2(-1)(+1) + (+1)(-1) = (-1) + (-1) + (-1)	-1
(-1,-1)	(+1, 0)	(-1)(+1) + 2(-1)(+1) + (+1)(0) = (-1) + (-1) + (0)	-1
(-1,-1)	(+1,+1)	(-1)(+1) + 2(-1)(+1) + (+1)(+1) = (-1) + (-1) + (+1)	u
(-1,-1)	(--, u)	(-1)(+1) + 2(-1)(--) + (+1)(u) = (-1) + (--) + (u)	u

Symbolic Computation of Concavity Con't

$(\mathcal{M}_i, \mathcal{C}_i)$	$(\mathcal{M}_j, \mathcal{C}_j)$	$(\mathcal{C}_i)(+1) + 2(\mathcal{M}_i)(\mathcal{M}_j) + (+1)(\mathcal{C}_j)$	Result
(-1, 0)	(-1,-1)	(0)(+1) + 2(-1)(-1) + (+1)(-1) = (0) + (+1) + (-1)	u
(-1, 0)	(-1, 0)	(0)(+1) + 2(-1)(-1) + (+1)(0) = (0) + (+1) + (0)	+1
(-1, 0)	(-1,+1)	(0)(+1) + 2(-1)(-1) + (+1)(+1) = (0) + (+1) + (+1)	+1
(-1, 0)	(0, 0)	(0)(+1) + 2(-1)(0) + (+1)(0) = (0) + (0) + (0)	0
(-1, 0)	(+1,-1)	(0)(+1) + 2(-1)(+1) + (+1)(-1) = (0) + (-1) + (-1)	-1
(-1, 0)	(+1, 0)	(0)(+1) + 2(-1)(+1) + (+1)(0) = (0) + (-1) + (0)	-1
(-1, 0)	(+1,+1)	(0)(+1) + 2(-1)(+1) + (+1)(+1) = (0) + (-1) + (+1)	u
(-1, 0)	(--, u)	(0)(+1) + 2(-1)(--) + (+1)(u) = (0) + (--) + (u)	u
(-1,+1)	(-1,-1)	(+1)(+1) + 2(-1)(-1) + (+1)(-1) = (+1) + (+1) + (-1)	u
(-1,+1)	(-1, 0)	(+1)(+1) + 2(-1)(-1) + (+1)(0) = (+1) + (+1) + (0)	+1
(-1,+1)	(-1,+1)	(+1)(+1) + 2(-1)(-1) + (+1)(+1) = (+1) + (+1) + (+1)	+1
(-1,+1)	(0, 0)	(+1)(+1) + 2(-1)(0) + (+1)(0) = (+1) + (0) + (0)	+1
(-1,+1)	(+1,-1)	(+1)(+1) + 2(-1)(+1) + (+1)(-1) = (+1) + (-1) + (-1)	u
(-1,+1)	(+1, 0)	(+1)(+1) + 2(-1)(+1) + (+1)(0) = (+1) + (-1) + (0)	u
(-1,+1)	(+1,+1)	(+1)(+1) + 2(-1)(+1) + (+1)(+1) = (+1) + (-1) + (+1)	u
(-1,+1)	(--, u)	(+1)(+1) + 2(-1)(--) + (+1)(u) = (+1) + (--) + (u)	u
(0, 0)	(-1,-1)	(0)(+1) + 2(0)(-1) + (+1)(-1) = (0) + (0) + (-1)	-1
(0, 0)	(-1, 0)	(0)(+1) + 2(0)(-1) + (+1)(0) = (0) + (0) + (0)	0
(0, 0)	(-1,+1)	(0)(+1) + 2(0)(-1) + (+1)(+1) = (0) + (0) + (+1)	+1
(0, 0)	(0, 0)	(0)(+1) + 2(0)(0) + (+1)(0) = (0) + (0) + (0)	0
(0, 0)	(+1,-1)	(0)(+1) + 2(0)(+1) + (+1)(-1) = (0) + (0) + (-1)	-1
(0, 0)	(+1, 0)	(0)(+1) + 2(0)(+1) + (+1)(0) = (0) + (0) + (0)	0
(0, 0)	(+1,+1)	(0)(+1) + 2(0)(+1) + (+1)(+1) = (0) + (0) + (+1)	+1
(0, 0)	(--, u)	(0)(+1) + 2(0)(--) + (+1)(u) = (0) + (0) + (u)	u

Symbolic Computation of Concavity Con't

$(\mathcal{M}_i, \mathcal{C}_i)$	$(\mathcal{M}_j, \mathcal{C}_j)$	$(\mathcal{C}_i)(+1) + 2(\mathcal{M}_i)(\mathcal{M}_j) + (+1)(\mathcal{C}_j)$	Result
(+1,-1)	(-1,-1)	(-1)(+1) + 2(+1)(-1) + (+1)(-1) = (-1) + (-1) + (-1)	-1
(+1,-1)	(-1, 0)	(-1)(+1) + 2(+1)(-1) + (+1)(0) = (-1) + (-1) + (0)	-1
(+1,-1)	(-1,+1)	(-1)(+1) + 2(+1)(-1) + (+1)(+1) = (-1) + (-1) + (+1)	u
(+1,-1)	(0, 0)	(-1)(+1) + 2(+1)(0) + (+1)(0) = (-1) + (0) + (0)	-1
(+1,-1)	(+1,-1)	(-1)(+1) + 2(+1)(+1) + (+1)(-1) = (-1) + (+1) + (-1)	u
(+1,-1)	(+1, 0)	(-1)(+1) + 2(+1)(+1) + (+1)(0) = (-1) + (+1) + (0)	u
(+1,-1)	(+1,+1)	(-1)(+1) + 2(+1)(+1) + (+1)(+1) = (-1) + (+1) + (+1)	u
(+1,-1)	(_, u)	(-1)(+1) + 2(+1)(_) + (+1)(u) = (-1) + (_) + (u)	u
(+1, 0)	(-1,-1)	(0)(+1) + 2(+1)(-1) + (+1)(-1) = (0) + (-1) + (-1)	-1
(+1, 0)	(-1, 0)	(0)(+1) + 2(+1)(-1) + (+1)(0) = (0) + (-1) + (0)	-1
(+1, 0)	(-1,+1)	(0)(+1) + 2(+1)(-1) + (+1)(+1) = (0) + (-1) + (+1)	u
(+1, 0)	(0, 0)	(0)(+1) + 2(+1)(0) + (+1)(0) = (0) + (0) + (0)	0
(+1, 0)	(+1,-1)	(0)(+1) + 2(+1)(+1) + (+1)(-1) = (0) + (+1) + (-1)	u
(+1, 0)	(+1, 0)	(0)(+1) + 2(+1)(+1) + (+1)(0) = (0) + (+1) + (0)	+1
(+1, 0)	(+1,+1)	(0)(+1) + 2(+1)(+1) + (+1)(+1) = (0) + (+1) + (+1)	+1
(+1, 0)	(_, u)	(0)(+1) + 2(+1)(_) + (+1)(u) = (0) + (_) + (u)	u
(+1,+1)	(-1,-1)	(+1)(+1) + 2(+1)(-1) + (+1)(-1) = (+1) + (-1) + (-1)	u
(+1,+1)	(-1, 0)	(+1)(+1) + 2(+1)(-1) + (+1)(0) = (+1) + (-1) + (0)	u
(+1,+1)	(-1,+1)	(+1)(+1) + 2(+1)(-1) + (+1)(+1) = (+1) + (-1) + (+1)	u
(+1,+1)	(0, 0)	(+1)(+1) + 2(+1)(0) + (+1)(0) = (+1) + (0) + (0)	+1
(+1,+1)	(+1,-1)	(+1)(+1) + 2(+1)(+1) + (+1)(-1) = (+1) + (+1) + (-1)	u
(+1,+1)	(+1, 0)	(+1)(+1) + 2(+1)(+1) + (+1)(0) = (+1) + (+1) + (0)	+1
(+1,+1)	(+1,+1)	(+1)(+1) + 2(+1)(+1) + (+1)(+1) = (+1) + (+1) + (+1)	+1
(+1,+1)	(_, u)	(+1)(+1) + 2(+1)(_) + (+1)(u) = (+1) + (_) + (u)	u

Appendix B. Proofs Associated with the DMC Transform

Proof of Arithmetic Shift Invariance

The proof of arithmetic shift invariance relies upon the definition of transform space addition (see Section 4.2.1). Specifically, the addition of any *DMC* time series signature and an encoded constant results in the identical time series signature. This type of addition defines the unique additive identity for *DMC*-space addition. Therefore, the proof simplifies as follows.

$$\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a} + c\mathbf{1}) = \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a} + \bar{c})$$

$$\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) + \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{c}) \qquad \textit{Definition of Additive Identity}$$

$$\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a})$$

Proof of Scale Invariance

The proof of scale invariance relies upon the definition of transform space multiplication (see Section 4.2.2). Similarly to addition, the multiplication of any *DMC* time series signature and an encoded constant results in the identical time series signature. This type of multiplication defines the unique multiplicative identity for *DMC*-space multiplication. Therefore, the proof again simplifies as follows.

$$\mathcal{Q}_{\overline{\mathrm{T}}}(c\bar{a})$$

$$\mathcal{Q}_{\overline{\mathrm{T}}}(c\mathbf{1}) * \mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a}) \quad \textit{Definition of Multiplicative Identity}$$

$$\mathcal{Q}_{\overline{\mathrm{T}}}(\bar{a})$$

Vita

David Conrad w[redacted]. He was award a Bachelor of Science in Computer Science from the United States Air Force Academy in 1991. Upon his commission, he was assigned to the 84th Test Squadron at Tyndall AFB, Florida where he instrumented U.S. Air Force fighter aircraft radar systems for operational testing and evaluation. He arrived at the Air Force Institute of Technology in May 1995 to pursue his Master of Science in Computer Engineering.

[redacted]